SEMANTIC MATERIAL LABELING OF 3D POINT CLOUDS USING RGB-D DATA AND VISUAL SLAM

by

SIVA KRISHNA RAVIPATI

(Under the Direction of Ramviyas N. Parasuraman)

ABSTRACT

The navigation ability and the perception of the objects in the environment are the basis for the mobile robots to explore an unknown environment. Understanding the surroundings and determining the objects in the environment of one of the major tasks in this regard. For this purpose, we propose a method with RGB-D sensor-based material mapping with simultaneous localization and mapping (SLAM) fusion. This method integrates the material classification network and SLAM to obtain the semantic map. This enables the robot to explore an unknown environment autonomously and identify the objects and materials in the environment. The material recognition or classification is done based on the images taken from the RGB-D camera, and SLAM is done using ORB_SLAM2. The RGB and Depth features are used for material classification, and we also explored different fusion methods, late fusion, and complementarity-aware (CA) fusion. Experiments are conducted to demonstrate this mechanism with a mobile robot, and an RGB-D camera installed, exploring an unknown environment consisting of objects of different material types. The results show that the mobile robot can successfully navigate the area by classifying and creating the material map.

INDEX WORDS: [Material Classification, Mobile Robot, Deep Convolutional Neural Networks, Simultaneous Localization and Mapping (SLAM), RGB and Depth features]

SEMANTIC MATERIAL LABELING OF 3D POINT CLOUDS USING RGB-D DATA AND VISUAL SLAM

by

SIVA KRISHNA RAVIPATI

M.Tech., SRM Institute of Science and Technology, India

A Thesis Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the Degree.

MASTER OF SCIENCE

ATHENS, GEORGIA

2023

SEMANTIC MATERIAL LABELING OF 3D POINT CLOUDS USING RGB-D DATA AND VISUAL SLAM

by

SIVA KRISHNA RAVIPATI

Major Professor:	Ramviyas N.Parasuraman
Committee:	Khaled Rasheed Suchendra M.Bhandarkar

Electronic Version Approved:

Ron Walcott
Dean of the Graduate School
The University of Georgia
May 2023

DEDICATION

To Mom and Dad, my wife, lovely daughters Hanvika and Parnika, and to my brother for always loving and supporting me.

ACKNOWLEDGMENTS

Firstly, I would like to extend my utmost gratitude to my principal advisor, Prof. Dr.Ramviyas N. Parasuraman, for giving me the opportunity to work with him and for his in-depth guidance on this research. During this work, he has provided me with many insightful suggestions and discussions, shared exciting topics, and provided the right direction for me. I would not have been able to finish this thesis without his supervision. I am sure that the experience of working with him would have a significant impact on my future career.

I am also grateful and thank my committee members, Dr. Khaled Rasheed and Dr.Suchendra M. Bhandarkar. I learned a lot of professional skills and new knowledge in their classes. Thanks for their kind and encouraging comments and considerate attitudes throughout the whole time means very much to me.

It has also been fantastic to work with several generous and supportive Collaborators. I want to thank Ehsan Latif, who kindly and patiently assisted me in the Lab. I also would like to thank the intelligent and cordial bunch of people in the HeRo lab, all folks of the Institute of Artificial Intelligence, and The University of Georgia, for their great help during the uniquely hard times of COVID-19.

Finally, I am so grateful to my family, wife Sarada, for their support and love over the years. My sincere and special thanks to my brother, Hari, for guiding and helping me through the most challenging situations in my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

We need robots to perform more advanced actions and exhibit human-like behaviors. To achieve this, we need them to reason and understand the world in a human-like fashion. In general, robots do not show autonomous intelligence. Except for basic control flow, they are mostly unaware of the environment and only limited to sensory input information during their operations. This almost complete lack of autonomy is one of the significant obstacles that must be overcome to allow robots to become mobile in a priori unknown environments.

This thesis describes our research undertaken to help mobile robots conquer, understand, and navigate indoor environments. When a robot is deployed in an environment such as a home or an office space, it is usually not feasible to equip it with an accurate model of that environment a priori. Therefore, the robot will first need to create a model of the world. A mobile robot, in general, needs to comprise a map that allows it to localize and plan a collision-free path according to its assignment. Building such a map from onboard sensors is a challenging problem, as the robot needs to be localized to create a map and localize itself. Therefore, these tasks need to be done concurrently, and the problem is referred to as simultaneous localization and mapping (SLAM). The SLAM problem has been subject to intensive research, and feasible solutions have been developed for many scenarios. Various visual SLAM algorithms have been developed and used for this purpose. However, these algorithms lack semantic information, are less accurate, and are of low speed. We direct our work in this area and develop a material semantic map of the environment by identifying the objects and determining their material type.

Material Recognition is a famous yet complex problem in computer vision. Everyday scenes contain various visually similar yet structurally different materials that are useful to identify. For example, autonomous robots and self-driving vehicles must be aware of whether they are driving on concrete, metal, pavement, or black ice. As further advances in robotics and human-computer interaction are made, the need for more accurate material recognition will grow. People recognize and categorize objects based on the visual information collected from the human vision system and prior knowledge of object classes. Diverse objects in the real world can be effectively characterized by their shape, color, and material type. However, most computer vision methods extract only color and shape information. Material recognition of an arbitrary object at a distance can improve the performance of conventional object recognition and scene understanding tasks. Extended modality of the reconstructed 3D model with material type information can augment touch-based interaction and realistic rendering in virtual and augmented reality applications. The challenge is to use a depth map and retrieve features from it to predict the type of material from a camera image.

In recent years, the usage of mobile or service robots in exploration and mapping applications has gained great interest and advancements. Mobile robots can be used in a variety of situations such as domestic aid, firefighting aids, service and logistics, and more. Nevertheless, various unsolved problems remain when deploying robots in unknown environments. In particular, it is critical for the mobile robot to perceive and learn the properties of the objects in an unknown environment to increase its autonomy and effectively perform its tasks. For example, to explore and navigate in new surroundings, it is essential to know the location of obstacles, doors, and other accessible areas to preplan the operation. To address this problem, researchers performed studies using several sensing modalities and machine learning algorithms to classify different objects and their material types. Of these modalities, computer vision has been prominently used because of the recent advances in deep learning algorithms and the availability of large datasets. There are several robots in use that depend on image processing and vision-based techniques; however, they perform poorly when the lighting conditions are bad or the environment is foggy which vastly reduces the visibility of the surroundings.

In addition, an ability to recognize different kinds of materials could play a significant role in robotics navigating and exploring applications. Understanding material properties (e.g. type, surface roughness, or deformability) of objects can be used to inform grasp planning and manipulation. Mobile robots should understand their surrounding materials when planning movements through precarious rubble under a collapsed building. Understanding the type of materials that make up the surroundings in an environment is a critical aspect of characterization and will help the robots in performing their actions like exploring, manipulating, and clearing., in an unknown environment.

§ 1.1 Motivation

Utilizing RGB-D images and Visual SLAM to create a semantic material map is an important area of research. It permits precise portrayal and comprehension of the materials and items in a scene. This information can greatly help robots in navigation and object manipulation and aid them in improving their performance. Applications, like augmented reality, may also benefit from this information.

The ability to precisely identify and classify the materials in a scene is one of the main benefits of creating a semantic material map. The ability to accurately identify the materials of objects is essential for safe and effective manipulation, and this semantic information can be used to improve performance in tasks like object manipulation. Additionally, the robot's ability to interact with its environment can be enhanced by accurately classifying materials, which can improve its navigational performance. The semantic material map can also be used by an augmented reality application to more accurately and realistically overlay virtual objects onto the real world.

Figure 1 depicts the singular outcome of our framework. The ORB SLAM2 point cloud, the object that was found, and their respective semantic material maps are all shown in the illustration. The integration of our framework's abilities to precisely identify objects and assign them relevant semantic information is highlighted in this display. The outcome depicted in Figure 1 demonstrates the efficacy of our framework in enhancing environmental perception through semantic mapping.

The capacity to produce a comprehensive point cloud map of the environment is yet another advantage of creating a semantic material map. Using a set of points in space, point cloud mapping creates a three-dimensional representation of the environment. The point cloud map can be utilized for a variety of purposes, including object manipulation, localization, and navigation. Additionally, the point cloud map has the ability to produce 3D models of the environment, which can be utilized in augmented reality applications as well as floor plans and building plans.

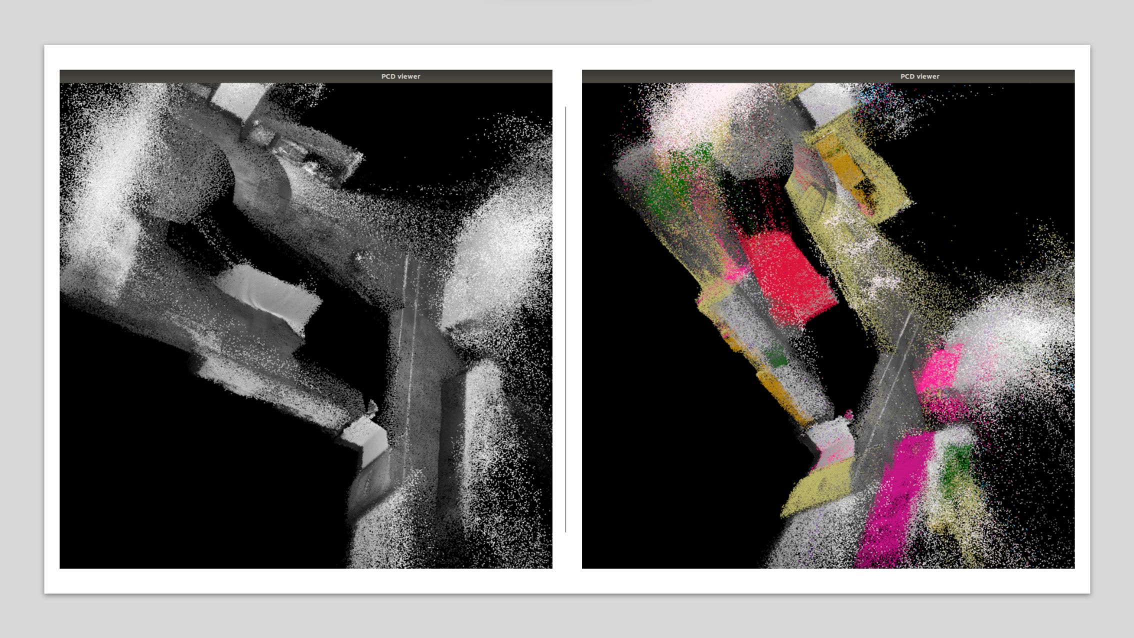


Figure 1.1: Example Semantic Material Labeling of the Point Cloud with Material Types indicated by different colors

Creating a point cloud map with the ORB-SLAM2 algorithm is also advantageous. The cutting-edge visual odometry and SLAM algorithm known as ORB-SLAM2 is highly accurate and effective. On a standard computer, it creates a global, consistent map of the environment in real-time. This makes it possible to create point cloud maps of high quality in real-time, which are necessary for numerous applications like robot navigation and augmented reality.

Using RGB-D images to create a semantic material map has a number of other advantages in addition to these. For instance, including depth information in RGB-D images enhances object detection and localization accuracy and provides additional information regarding the scene's objects' shape and texture. Using RGB-D images can also make the system more resistant to changes in lighting and give a more accurate picture of the environment in low light.

In conclusion, using RGB-D images to create a semantic material map makes it possible to accurately depict and comprehend the materials and objects in a scene. Various applications, including augmented reality, robot navigation, and object manipulation, may benefit from this information's enhanced performance. With RGB-D images, object detection and point cloud mapping can offer the following benefits:

- a detailed and rich depiction of the environment.
- resulting in improved performance across a variety of tasks, including localization.
- Navigation.
- manipulating objects

Unfortunately, most works have addressed the problems of detecting and classifying materials in a single RGB image, and few others have used the Depth image, but do not use the combined and complementary information to perform material mapping in an environment.

In short, this article has done the main works as follows:

- We explore the usage of RGB and Depth data for material classification of objects and different fusion techniques of these features.
- We use RGB-D data, Yolov5 for object detection, and ORB SLAM2 to build a point cloud map directly, reducing the cost and complexity of the system.
- We propose to use RGB-D data, the point cloud, and the MSCC algorithm for material mapping with SLAM for robot navigation.
- We generate a semantic map that allows for a more detailed understanding of the environment, as it not only provides semantic material information but also the object information in the environment.

This paper is organized as follows. In Chapter 2, the related works about object detection, material recognition, multimodal feature learning from RGB-D images, and point cloud segmentation are discussed. In Chapter 3, the methodology and fusion techniques explored are presented. In Chapter 4, the experiments conducted are discussed and analyzed. More insights about this method are also presented in this section. Lastly, in Chapter 5, the conclusion of this work is drawn.

CHAPTER 2

RELATED WORK

In this section, we will review the previous works based on fives aspects: 2D Object detection, RGB-D multi-modal feature learning, Material recognition, Visual SLAM, and 3D point cloud segmentation and analysis methods. We firstly review 2D object dection in Section 1.1, RGB-D multi-modal feature learning in Section 1.2, material recognition in Section 1.3, Visual SLAM in Section 1.4 and 3D point cloud segmentation and analysis in Section 1.5. Finally, we will give a discussion in Section 1.6.

§ 2.1 2D Object Detection

This section discusses the progress made in 2D object detection by adopting deep learning based on monolithic and two-stage architectures. The monolithic architectures are a single-stage object detection pipeline and two-stage architectures comprised of region proposals followed by the classification, regression, and other post-processing.

One of the earliest monolithic architectures that incorporated deep learning to predict the category and precise location of objects within an image was the work on DetectorNet by Szegedy et al. Szegedy et al., 2013. DetectorNet, built upon the AlexNet Krizhevsky et al., 2017 architecture, specializes in the detection task and needs to be trained separately for every object category and mask type, thereby incurring substantial computational complexity. Another famous work published was OverFeat, proposed by Sermanet et al. Sermanet et al., 2013. This network is used to simultaneously learn the detection of objects, classification, and localization and is much speedier than DetectorNet.

Liu et al. W. Liu et al., 2016 presented Single-shot Multibox Detector, which uses VGG-16 architecture as a backbone as it performed the best in image classification and transfer learning tasks then. It is a single-shot object detection model that uses multi-scale feature maps and convolutional kernels for detection. SSD obtained high accuracy with highly-improved computation speed. Later, researchers came up with Feature Fusion SSD (FSSD) Z. Li and Zhou, 2017, an improvement on top of SSD to include a lightweight add-on module for fusing features. This method increased accuracy but decreased detection speed compared to SSD.

Another significant breakthrough in single-shot 2D object detection came through Redmon et al. Redmon et al., 2016. This model, entitled You Only Look Once (YOLO), was not a classifier disguised and modified for object detection tasks and could detect objects at real-time speed. But, YOLO could not predict small objects in close proximity as each cell can only have two bounding box predictions of a single class. Many improvements have been made in the subsequent years. One such approach was YOLOv2 Redmon and Farhadi, 2017, which made changes like introducing batch normalization, increasing input resolution, predicting multiple objects per grid cell with anchors, etc. This architecture was further improved upon in Fast YOLO Shafiee et al., 2017, where motion-adaptive inference was employed to attain even faster detection with a more compact architecture. In YOLOv3 Redmon and Farhadi, 2018, the objectness score for each bounding box by logistic regression is computed and used a relatively larger architecture, DarkNet-19, with residual shortcut connections. Later, YOLOv4 Bochkovskiy et al., 2020 and YOLOv5, faster alternatives, weres introduced by incorporating bag-of-freebies and bag-of-freebies techniques during the training phase.

In region proposal-based architectures, object detection is done in two stages. In the first stage, multiple region proposals are generated from the image. After that, in the second stage, features are extracted from these proposals using a ConvNet and, thereby, classify them into object categories. The Region-based Convolutional Neural Networks (R-CNN), proposed by Girshick et al. Girshick et al., 2014, was a notable work in the history of CNN-based object detection. However, one of its significant disadvantages was its computation time complexity. Performing a selective search on each input image to generate 2̃000 region proposals and passing them through CNN to generate features is time-consuming. Later, Girshick Girshick, 2015 improved upon R-CNN and proposed Fast R-CNN. Another subsequent improvement by Ren et al. Ren et al., 2015 is through Faster R-CNN, where there was no separate selective search step to generate object proposals. The weights of the CNN are shared between region proposal and object detection tasks, and the process is even faster than earlier. An extension on top of Faster R-CNN, Mask R-CNN, was introduced by He et al. He et al., 2017. In this, pixel-level segmentation of each region of interest is done along with object detection.

§ 2.2 RGB-D multi-modal feature learning

Multi-modal feature learning has been investigated in various works, J. Yu et al., 2014 proposed a multimodal hypergraph learning based sparse coding method for the click prediction. Multimodal distance metric learning based method was proposed in J. Yu et al., 2016. A promising Multiview Hessian regularization algorithm was proposed by W. Liu and Tao, 2013 to combine multi-modal features for semi-supervised learning. Various strategies have been proposed to fuse the multi-modal features effectively in Q. Wang et al., 2020. These fusion methods are divided into four categories. 1) Feature-level multi-modal combination. S. Song et al., 2015 concatenated the two-stream CNN features and fed them into fully connected layer to fuse the multi-modal features. In X. Song et al., 2017, the authors exploited a three stream CNN to combine RGB features and two branches of depth features with element-wise summation. 2) Image level multi-modal fusion. In Couprie et al., 2013, the authors constructed the RGB-D Laplacian pyramid with the RGB modality and depth modality. 3) Modal correlative feature fusion. Modal consistent features are learnt in A. Wang et al., 2015 by enforcing the RGB features to be close to the depth features. 4) Modal correlative and distinctive features fusion. The authors of Y. Li, Zhang, et al., 2018 attempted to learn the correlative and distinctive features simultaneously between the RGB and depth modality. The multi-modal feature fusion methods ignore the flexible local features and nevertheless, enforcing the consistency between multiple modalities obstructs the model from learning the modal complementary features.

§ 2.3 Material Recognition

Materials recognition is a challenging task due to wide variation in appearance within categories. Previous research work on material recognition predominantly focused on material classification based on 2D RGB images, and minimal work based on depth features. Majority of the earlier works have concentrated on variance of oriented gradientsD. Hu et al., 2011, reflectance based edge featuresLee et al., 2022; Lee et al., 2019; C. Liu et al., 2010, and pairwise local binary patternsX. Qi et al., 2014. In recent years, CNN features Schwartz and Nishino, 2013Cimpoi et al., 2013Kalliatakis et al., 2017 have been employed to achieve the state of the results in material classification on many public material datasets. In some works DeGol et al., 2016, the 2D features such as texture and color are combined with 3D geometry features like surface normal, camera intrinsic and extrinsic parameters to improve material classification. For pixel wise material segmentation, Bell et al., 2014 convert patch based trained CNN classifiers into efficient fully convolutional framework that is combined with a fully connected CRF for performing pixel wise material recognition. Superior result are achieved Schwartz and Nishino, 2016 by combining local appearance with separately recognized global context cues including objects and places. The authors employed fully convolutional network (FCN) Long et al., 2014 followed by recurrent neural network(RNN) for dense pixelwise material segmentation. A novel CNN architecture is proposed T. Wang et al., 2016 by training on 4D light-field images and employing FCN for per-pixel material recognition. The work by Cheng Zhao et al, Zhao et al., 2017 is interesting and close to our work but they used 2D images for material recognition. In Kim et al., 2018, surface roughness is estimated from depth images for material classification, but this work did not employ deep learning algorithms as large dataset is not available. However, none of these methods use depth features of RGBD images combined with RGB features and employ CA fusion strategy for material recognition.

§ 2.4 RGB-D SLAM and Semantic SLAM

Over the last few years, direct and indirect methods have been used extensively in visual SLAM and traditional simultaneous localization and mapping (SLAM) systems. Getting proper depth of features and objects in the camera's view can significantly help the SLAM systems' performance. There are various ways to obtain depth information, and it can mainly be obtained using three types of cameras, stereo cameras, structured light cameras, and time of flight (ToF) cameras. Using the high-resolution depth map, visual SLAM increased their performance, both in accuracy and computational load. Some notable works on this front are RGBD SLAM v2 by F.Endres et al. Endres et al., 2014, ORB SLAM Mur-Artal et al., 2015 and its improved version ORB SLAM2 Mur-Artal and Tardós, 2017 and the latest ORB SLAM3 Campos et al., 2021. Other significant works in RGBD SLAM include Bundle Fusion Dai et al., 2017 and BAD SLAM Schops et al., 2019

Also, the existing SLAM systems mainly focus on using low-level geometric features, such as points, lines, and planes, which cannot provide semantic information. Semantic SLAM can give semantic details on environments. This information help robots to understand surrounding scenes at both geometrical and content level. SLAM++ Salas-Moreno et al., 2013, proposed by Salas-Moreno et al., performs object detection in RGB-D tracking and mapping. The research work by John McCormac et al. on semantic SLAM was SemanticFusion McCormac et al., 2017. The method comprises a convolution neural network to produce class probability maps and fuse predictions into the 3D map. Keisuke Tateno et al. proposed a real-time dense monocular CNN-SLAM Tateno et al., 2017. With the aid of CNN, this SLAM can perform depth prediction and semantic segmentation. DA-RNN Xiang and Fox, 2017 introduced a new recurrent neural network (RNN) architecture for semantic labeling on RGB-D videos, which utilizes information from multiple viewpoints to improve segmentation performance. DS-SLAM C. Yu et al., 2018 and SDF-SLAM Cui and Ma, 2020 are other interesting works done for indoor environments.

§ 2.5 3D Point Cloud Analysis and Segmentation

Point cloud segmentation divides point clouds into different segments, each of which has similar properties. This process is a fundamental step towards scene understanding from point clouds. Driven by specific applications, 3D point cloud segmentation has become a very active research topic. Point Cloud Library (PCL) Rusu and Cousins, 2011 is a popular library that provides open-source segmentation algorithms. The early approach Schnabel et al., 2007 used RANSAC to detect planes from the point clouds, and then it divides objects with Euclidean separation. Adams et al. Adams and Bischof, 1994 proposed a region-growing algorithm in 2D image processing work. Later, this algorithm was used in the works related to the 3D point cloud. Based on the smoothness feature, Rabbani et al. Rabbani et al., 2006 presented a method for segmenting point clouds that finds smoothly connected areas in point clouds.

The Locally Convex Connected Patches (LCCP) algorithm by Stein et al. Stein et al., 2014 uses a normal vector to judge local convexity, which can help divide the point cloud into some individual segments. Later for both automatic and interactive segmentation, Golovinskiy et al. Golovinskiy and Funkhouser, 2009 proposed a min-cut-based method of segmenting objects in point clouds. 3D point clouds are irregular and unordered, unlike 2D images. Thus, the standard way in 2D image processing, like convolution, cannot be used for 3D point clouds. Deep neural network-based methods have recently been proposed for 3D point cloud classification and segmentation, such as PointNet C. R. Qi, Su, et al., 2017 and PointNet++ C. R. Qi, Yi, et al., 2017. The PointNet was able to learn directly from unordered point clouds, which combine local point features and global information to perform 3D segmentation. Based on PointNet, PointNet++ introduced a hierarchical neural network to learn local features with increasing contextual scales, which can learn deep point set features efficiently and robustly. DGCNN Wu et al., 2018 uses an EdgeConv suitable for high-level tasks on the point cloud. PointCNN Y. Li, Bu, et al., 2018 presents a novel X-transformation approach, which can take advantage of CNNs for point cloud processing. Randla-net Q. Hu et al., 2020 introduced a lightweight neural architecture that can process large-scale point clouds. This network is 200 times faster than other architectures like PointNet and PointCNN. However, these methods only use point cloud information and are difficult to extend to semantic labeling.

§ 2.6 Discussion

In summary, previous material recognition methods have focused on 2D RGB images and few have combined depth features, and have not used complementary information. Also, the semantic slams concentrated their object detections in the environment. Predominantly, other multimodal learning used other modalities, such as tactile and acoustic sensing, surface reflectance, and roughness, and has largely employed machine learning algorithms. To most of our knowledge, this work is the first for learning the modal complementary information from RGB and depth features for material classification or recognition. Also, this work is one of a kind that allows for a more detailed understanding of the environment, as it not only provides semantic information but also material information.

CHAPTER 3

METHODOLOGY

§ 3.1 Approach

Our system architecture consists of 4 components: The object detection component, the RGB-D-based material classification network, visual SLAM component, and the voxel-based matching component. The overview of the system is shown in Figure 3.1

Object Detection: In this module, the objects are detected within the RGB images captured by the camera using the state-of-the-art Yolov5 network. The network outputs a set of 2D bounding boxes, each with a class label and a confidence score. The class label provides information on what objects are present within the image, and the confidence score provides information on how confident the network is in its prediction.

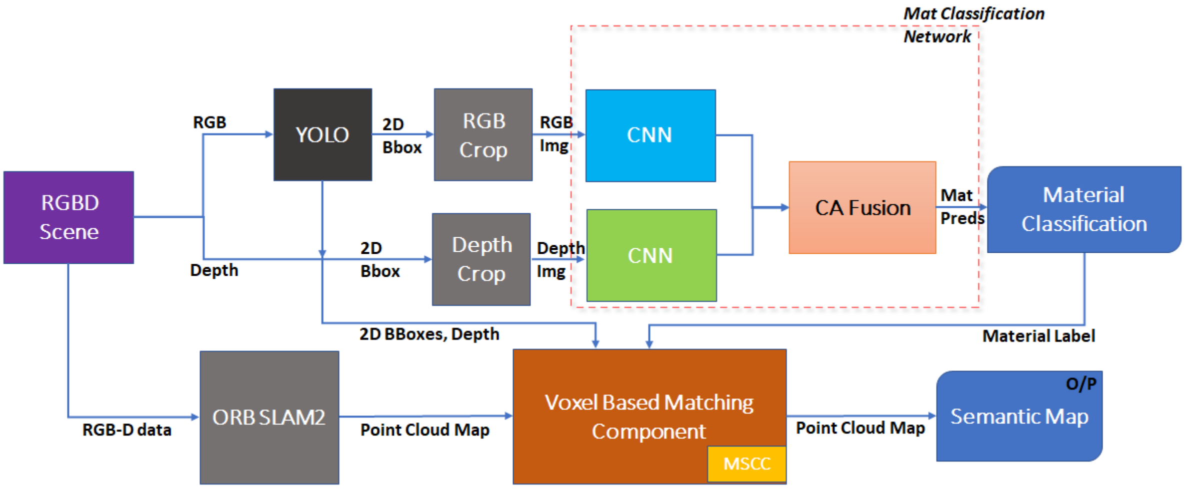


Figure 3.1: Overview of the system

Material Classification: Once the objects are detected, we use the detected object bounding boxes, crop the RGB frames, and crop the corresponding bounding boxes from the Depth frames. The cropped

RGB and Depth images are fed as input to the Material Classification Network module. This network module predicts the object's material, such as metal, plastic, etc.

Visual SLAM: We use ORB-SLAM2 Mur-Artal and Tardós, 2017 in this module. ORB-SLAM2 is a visual odometry and SLAM system that can be used with monocular, stereo, and RGB-D cameras. It is based on the ORB feature descriptor and uses bundle adjustment optimization to improve the accuracy of the point cloud map. ORB-SLAM2 is a real-time system that can handle large-scale and dynamic environments. It generates a sparse 3D point cloud map of the environment, which is used as a reference for the 3D coordinates of the environment.

Voxel-based matching component: The Voxel Based Matching Component is a crucial part of the pipeline. This component receives the point cloud map generated by ORB-SLAM2, the 2D bounding boxes obtained from YOLOv5, and material labels from the Material classification component as input. The voxel-based matching component uses depth information to map the 2D bounding boxes to 3D bounding boxes in the same coordinate system as the point cloud. This component matches the 3D bounding boxes obtained from object detection with the 3D coordinates of the point cloud. Later this component propagates the material labels to the points in the point cloud.

In summary, the visual SLAM (ORB-SLAM2) component generates the point cloud map, and the YOLOv5 component detects objects and locates the bounding boxes of the objects in the images. The material classification component classifies the objects in the bounding boxes into different material classes. The voxel-based matching component uses the point cloud map generated by ORB-SLAM2 and the material labels obtained from the material classification component to match the 3D coordinates of the bounding boxes with the 3D coordinates of the point cloud and propagate the material labels to the points in the point cloud.

The final semantic map is represented as a 3D point cloud, where each point in the point cloud is associated with both semantic and material labels.

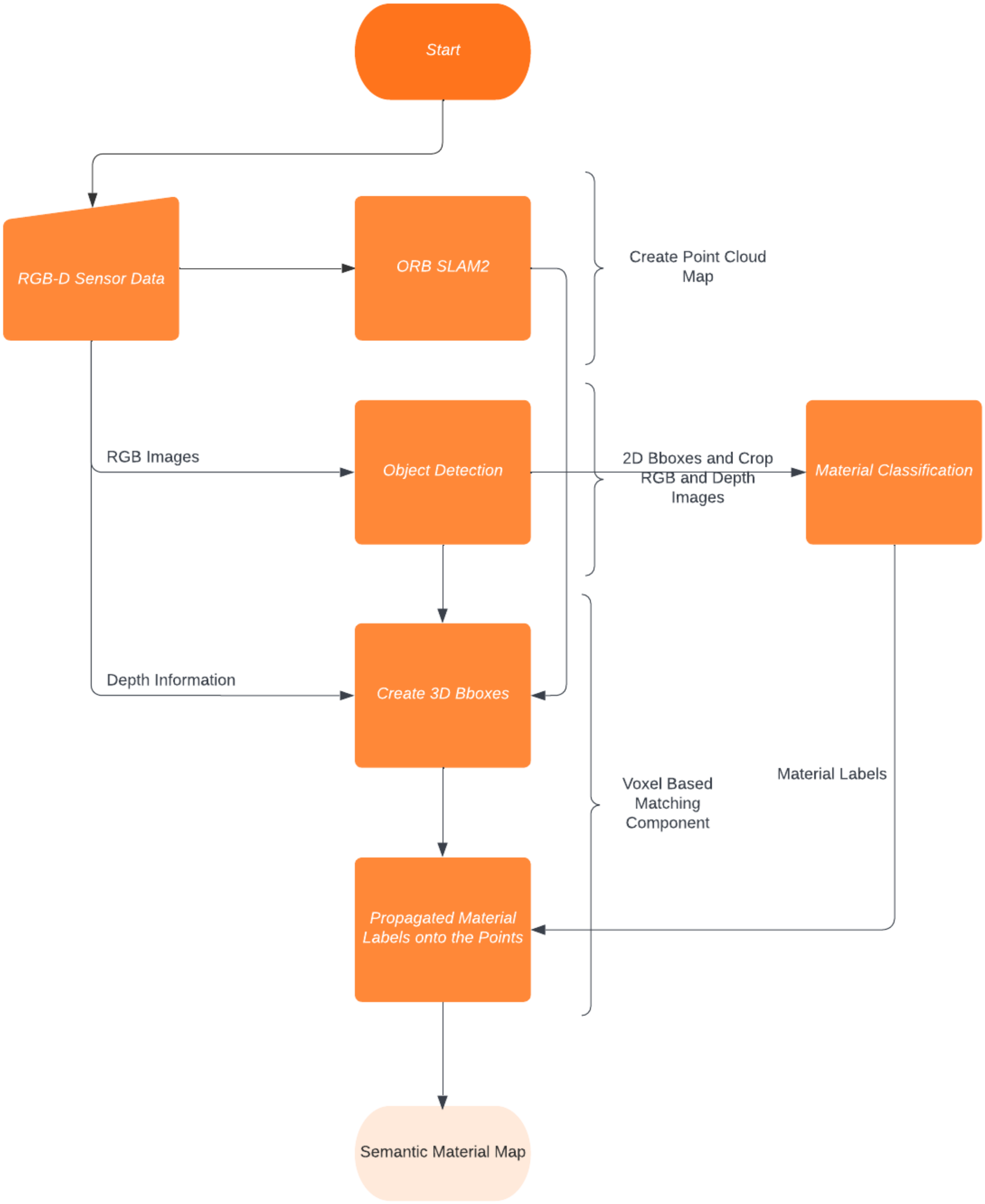


Figure 3.2: Flowchart of the system

§ 3.2 Object Detection using YOLOv5

The task of identifying and locating objects in images or videos is known as object detection. Over the course of time, a number of object detection models, such as R-CNN, Fast R-CNN, Faster R-CNN, and YOLO (You Only Look Once), have been developed. In order to predict the bounding boxes and class labels of the objects in the images and extract features from the input images, they make use of a combination of machine learning algorithms and convolutional neural networks (CNNs).

Jocher et al., 2022 created the cutting-edge object detection model known as YOLOv5 (You Only Look Once version 5). It is a system for detecting objects in real time that is quick, accurate, and adaptable. On a variety of benchmarks, including the COCO object detection dataset, this model has produced excellent results. YOLOv5 has consistently ranked among the best object detection models on this dataset, frequently outperforming other SOTA models in terms of accuracy and speed. The capacity of YOLOv5 to make predictions at a variety of scales is one of its main advantages. The model can process inputs at multiple scales and make predictions at multiple scales thanks to the use of a technique known as "feature pyramid networks." This procedure makes it possible for the model to distinguish between small and large objects in the same image—something that can be difficult for other object detection methods.

YOLOv5 is a newer version of YOLO (You Only Look Once) that is better at identifying objects in two-dimensional images. A fully convolutional neural network (CNN) is used to predict object bounding boxes and class probabilities directly from an input image in this anchor-based, single-stage object detector. A series of convolutional and max pooling layers, followed by several fully connected layers, make up YOLOv5's network structure. The YOLOv5 network structure can be summarized as follows:

Input Layer: YOLOv5 takes any RGB image as its input. In order to extract features from the image, it is processed through a series of convolutional and max pooling layers. The image is processed by the model in a grid of cells, with each cell predicting the class probabilities and bounding boxes for a group of nearby pixels. The size and shape of the objects in an image can be predicted using anchor boxes. Anchor box sizes that are closest to the objects in the training data are chosen by the YOLOv5 model through the use of k-means clustering.

Max pooling and convolutional layers: A series of 3x3 convolutional filters make up the first layer, which is followed by a 2x2 max pooling layer. The number of filters in this layer is followed by a succession of similar layers, with each one reducing the output tensor's size.

Connected Layers: The convolutional and max pooling layers of the network are followed by a series of fully connected (FC) layers that predict the bounding boxes and class probabilities for each image object by combining the extracted features. The shape of the final FC layer is (S, S, B*(5+C)), where S represents the number of cells in the grid, B represents the number of anchor boxes per cell, and C represents the number of object classes.

Output Layer: YOLOv5s produces a list of class probabilities and bounding boxes for each image object that is detected. Non-maximal suppression (NMS) preserves only the most reliable predictions by removing overlapping bounding boxes.

A number of models in YOLOv5 have been developed for a range of applications and trained on various datasets. One or more of these models is:

- YOLOv5l model is the lightweight version of YOLOv5. This model was made to be speedy and efficient. Because it contains fewer parameters and was trained on a smaller dataset, it can be used on devices with constrained resources.

- YOLOv5x model is a bigger and more accurate variant of YOLOv5 for high-performance object identification. Because it was trained on a larger dataset and contains more parameters, it is appropriate for situations where precision is more crucial.

- YOLOv5s model is comparable to YOLOv5l, despite being intended to be even faster and more efficient. It can be applied to cellphones or low-power embedded systems.

- YOLOv5m model, which is a medium-sized version of YOLOv5, strives for both efficiency and accuracy at the same time. Because it was trained on a medium-sized dataset and has a moderate number of parameters, it can be used for a wide range of tasks.

Each of these models can be optimized for different use cases, and in our application, we need a fast and efficient model that can run on a resource-constrained device. So we preferred to use the YOLOv5s model, which was trained on the COCO dataset.

§ 3.3 Material Classification Network

Many researchers have tried to recognize material types based on color features; however, the material type of an object is not entirely correlated with its visual appearance. Hence we try to evaluate using the other modalities to achieve better accuracy. Various works are available to incorporate different modalities; here, we try to use depth information and extract these from the RGB-D images.

The current material recognition literature includes using RGB-D images for feature extraction and classifying the material type. Varieties of strategies have been proposed to fuse the multi-modal features effectively like image level multi modal fusion, feature level multi modal fusion and modal correlative feature fusions. The complementarity-aware (CA) fusion module detects the objects available in the images using complementary features. The proposed method's novelty is extracting the distinctive and correlative features and fusing these features, similar to the CA fusion module. Also, previously the CA fusion module was used for salient object detection tasks.

Here, we explored two fusion techniques for material classification.

1. Late Fusion
2. Complemantarity-Aware (CA) Fusion

For RGB-D based material classification models, it is important to effectively fuse RGB images and depth maps. In this paper, we explore two fusion strategies, late fusion, and complementarity-aware(CA) multi-scale fusion.

§ 3.3.1 Late Fusion

Late fusion-based methods can be further divided into two categories: (i) two parallel networks adopted to learn high-level features for RGB and depth data, respectively, which are concatenated and then used to generate the final prediction Desingh et al., 2013; Han et al., 2018; N. Wang and Gong, 2019. This is called as later feature fusion. (ii) Two parallel networks are used to obtain independent classifications are done for RGB images and depth cues, and then the two saliency maps are concatenated to obtain a final prediction map Ding et al., 2019. This is called late result fusion.

In this method, the RGB image and the depth map are processed separately, i.e., to produce various types of features using two different networks . These features are later fused together, either by concatenation or by further processing using convolutional networks. The two networks feature maps output are then concatenated and presented to a final fusion network.

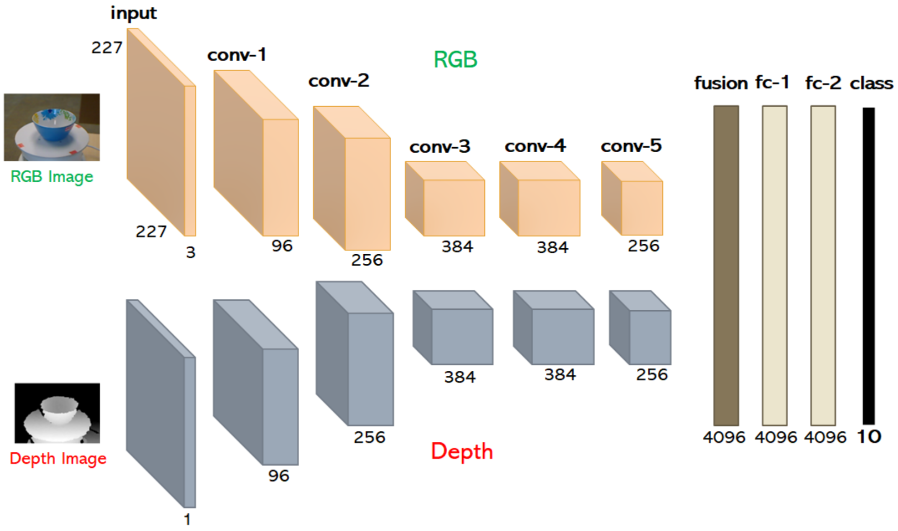


Figure 3.3: Architecture of Late Fusion CNN

§ 3.3.2 Complementarity-Aware(CA) Fusion

In this method, instead of using features learned separately from the color and depth modalities that yield some specific patterns, we also try to use the shared common modal patterns. For this we draw inspiration from the method proposed by Cheng et al. In the paper Chen and Li, 2018 the authors proposed a fusion mechanism called complementarity-aware (CA) fusion. This meachanism encourages the determination of complementary information from the different modalities at different abstraction levels. The authors introduced a CAFuse module, which enables cross-modal, cross-level connections and modal/levelwise supervisions, thereby the complementary information from the counterpart is captured. This reduces fusion ambiguity and increasing fusion efficiency.

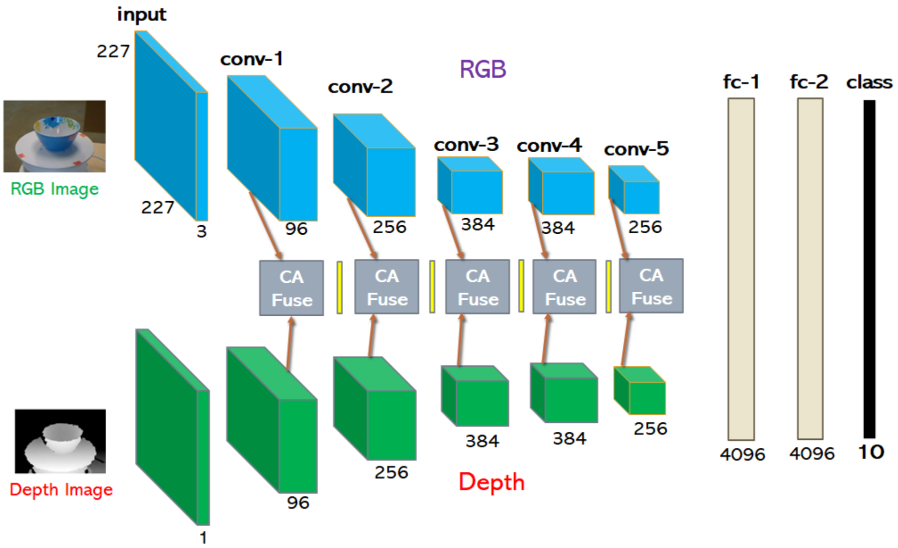


Figure 3.4: Architecture of CA-Fusion CNN

To excavate the complementarity of two modalities and remain the discriminability of cross-modal features, we use a Complementarity-aware Fusion Network (CAFN). We first model the usefulness of information from two modalities, then select complementary information of two modality features in spatial dimension with two symmetry gates, and finally a channel-wise weighting mechanism is conducted to fuse them for capturing more discriminative cross-modal features. The fused features retain not only information existing in both modalities, but also modality-specific information.

Because of the different image generation mechanisms between RGB and depth images, how to fuse cross-modal features effectively is a key issue for RGB-D based material classification.

CA-Fusion network (CAFN) includes two symmetric backbones for RGB and depth feature extraction and five cascaded Complementarity-aware fusion modules (CAF). We use ResNet-101 as unimodal symmetric backbones. We remove the average pooling and the fully connected layers of backbone and the last two stages are modified with dilated convolution to maintain feature resolution for keeping more spatial information. Then we use hierarchical features from RGB and depth branches respectively, i.e., $\{F^i_{rgb}$ | i=1,2,3,4,5} and $\{F^i_d$ | i=1,2,3,4,5}. CAFN can select complementary information from two modalities and then fuse enhanced unimodal features for accurate cross-modal features with the help of CAF. Two unimodal features $F^i_{rgb} \in R^{C_i \times H_i \times W_i}$ and $F^i_d \in R^{C_i \times H_i \times W_i}$ extracting from corresponding backbones are sent to CAF, where C_i, H_i and W_i refer to the channel, height, and width number of the i-th layer respectively.

§ 3.4 Visual SLAM

The point cloud map of the environment was obtained by employing a visual SLAM algorithm based on RGB-D. A technique known as RGB-D SLAM (RGB-D Simultaneous Localization and Mapping) is used to estimate a sensor's pose—that is, its position and orientation within the environment—in real time while simultaneously creating a 3D map of the environment. It is based on the idea of using an RGB camera to take color images of the environment and a depth sensor, such as a structured light sensor or a time-of-flight camera, to measure the distance to nearby surfaces.

Matching keypoints—distinct features—in the RGB images to keypoints in the depth images is how the RGB-D SLAM algorithm works. This information is then used to estimate the sensor's pose and create a 3D map of the environment. The 3D map and sensor pose estimate are continuously updated by means of a window of recent images that the algorithm employs using a sliding window approach.

Mur-Artal and Tardós, 2017 developed ORB-SLAM2, a tailored version of RGB-D SLAM. It uses a combination of point features and line features for robust and accurate localization and mapping and is based on the ORB (Oriented Fast and Rotated Brief) feature descriptor. ORB-SLAM2 is a SLAM (real-time simultaneous localization and mapping) system that can simultaneously reconstruct the environment and locate a monocular, stereo, or RGB-D camera in a 3D environment. The ORB-SLAM system, which was initially developed for monocular cameras but has since been expanded to handle stereo and RGB-D cameras as well, serves as the foundation for ORB-SLAM2.

We used the ORB-SLAM2 as it offers the following features:

- Real-time performance: ORB-SLAM2 runs in real-time, even on low-end hardware.
- Scalability: ORB-SLAM2 handles large-scale environments and can recover from long-term camera drifts.

- Robustness: ORB-SLAM2 handles dynamic scenes, scale and viewpoint changes, and can recover from temporary tracking failures.
- Multi-Camera Support: ORB-SLAM2 works with monocular, stereo, or RGB-D cameras, as well as multiple cameras at the same time.
- Portable: ORB-SLAM2 is written in C++ and has been tested on various platforms, such as Ubuntu, Windows, and Android.

The following steps are required to obtain the point cloud map using ORB-SLAM2:

Data collection: Using a single camera or a collection of cameras, we begin by acquiring the visual data—RGB, RGB-D, and stereo images—in the first step. The ORB-SLAM2 algorithm receives this data as an input.

Extracting attributes: We then take the visual data and extract features from it. The ORB (Oriented Fast and Rotated Brief) feature descriptor is used by the ORB-SLAM2 algorithm to extract features from images. These features are used to create a map of the environment and track the motion of the camera.

Estimated camera position: The ORB-SLAM2 uses visual odometry to estimate the camera's pose using the extracted features. The camera's movement in relation to the frame before it is estimated by this. The motion of the camera is accumulated in this step at each frame to determine the camera's precise position and orientation.

Local cartography: The estimated camera pose is used by the ORB-SLAM2 algorithm to create a sparse map of the environment. The algorithm optimizes the map by adjusting the positions of the features and the camera pose as the camera moves across the screen.

Example screenshots of the ORB-SLAM2 done on our custom dataset are shown in Figure 3.4

Detection of loop closure: A loop closure detection module is used by ORB-SLAM2 to determine when the camera revisits a previously visited area of the map. The algorithm performs a global bundle adjustment to optimize the entire map and correct any errors that may have accumulated over time when a loop closure is detected.

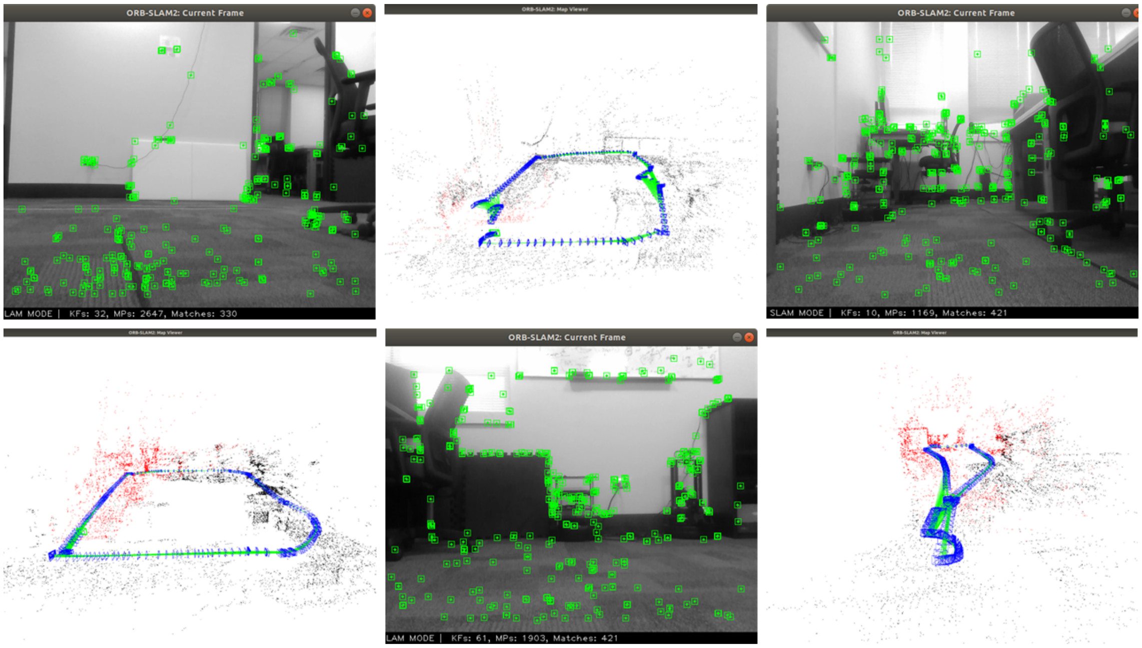


Figure 3.5: Example screenshots of ORB SLAM2 on our custom dataset

Making point cloud maps: By triangulating the 2D feature coordinates in the images with their corresponding 3D camera poses (derived from visual odometry and the bundle adjustment process), the point cloud map is created. Structure from Motion (SfM) is another name for this procedure. The final point cloud can be sparse or dense depending on the method used to generate it.

Post-processing: Outlier removal, noise reduction, and feature smoothing are just a few of the additional post-processing steps that are performed to enhance the image's quality.

§ 3.5 Voxel Based Matching Component

In this section, we discuss the crucial component of the system, the Voxel-Based Matching Component. Furthermore, using point cloud segmentation, the Multi-scale connected components algorithm in this component improves the accuracy of the material label propagation.

The inputs to the voxel-based matching component are:

- Point cloud map generated by ORB-SLAM2 component.

- 2D bounding boxes obtained from the YOLOv5 component.
- Material labels of the objects obtained from the material classification component.
- RGB-D sensor depth information.

The module incorporates depth information from the RGB-D sensor to calculate the 3D positioning of bounding boxes in the camera coordinate system. The voxel-based matching component employs a two-step process. Firstly, it utilizes 2D bounding boxes generated from YOLOv5 detections and the depth of each pixel within the bounding box to determine the 3D coordinates. The resulting 3D bounding boxes are in the same system as the point cloud created by ORB-SLAM2.

In order to match the 3D coordinates of the bounding boxes with the 3D coordinates of the point cloud, the module first partitions the point cloud into smaller units called voxels through a process known as voxelization. By associating each point in the point cloud with a voxel, the voxel-based matching component can efficiently determine the closest bounding box for each voxel. This reduction in the number of points being considered streamlines the processing of the point cloud for segmentation and material label propagation.

Following the creation of the voxel grid, point cloud segmentation is applied to separate the voxel grid into distinct clusters of points. In the subsequent section, the focus will shift to point cloud segmentation, exploring the Multi-scale connected components (MSCC) algorithm and its usage in the module.

§ 3.5.1 Point Cloud Segmentation

Point cloud segmentation is the process of dividing a 3D point cloud into smaller, semantically meaningful clusters. This enables the separate analysis and understanding of each object or surface within the point cloud. It is a helpful tool for tasks such as object recognition, scene understanding, and robot navigation, where it is essential to identify and classify different parts of a 3D scene.

There are many different techniques for point cloud segmentation, each with their own advantages and disadvantages. Some popular methods include region growing, region splitting, and graph-based methods. Region-growing methods start with an initial seed point and add neighboring points to the same cluster if they satisfy certain criteria, such as being within a certain distance of the seed point. Region-splitting methods divide the point cloud into multiple regions by iteratively splitting larger regions into smaller ones based on certain criteria, such as the variance of the points within the region. Graph-based methods construct a graph of the point cloud, where each point is represented by a node and edges connect neighboring points. The segmentation is then found by partitioning the graph into multiple connected components.

§ 3.5.2 Multi-scale connected components (MSCC) Algorithm

One of the most commonly used algorithms for point cloud segmentation is the Multi-Scale Connected Components algorithm. This algorithm is based on the work done by the authors Trevor et al., 2013 and Huang et al., 2022

This algorithm is a powerful method for segmenting point clouds by grouping points that are spatially close and similar to one another. The MSCC algorithm extends the mechanism of the classic connected component labeling algorithm, which is used to identify connected regions of pixels or voxels in a 3D image.

The Multi-Scale Connected Components (MSCC) algorithm segments the point cloud based on a multi-resolution approach. First, a voxel grid is created from the point cloud, where each voxel represents a small cube of space. Then, the algorithm looks for components that are connected. Connected components are groups of voxels that are connected to each other.

The MSCC algorithm separates objects or surfaces better than other segmentation techniques as it uses a multi-resolution approach. This means it can separate a point cloud at different scales, which can be useful for identifying difficult-to-distinguish objects or surfaces. Additionally, the MSCC algorithm extracts the shape of an object at different levels of detail.

The steps involved in Multi-Scale Connected Components (MSCC) algorithm are:

- Initialization: Set the scale parameter to the maximum scale and the minimum scale.

- Voxelization: Divide the point cloud into voxels based on the current scale.

- Connected Component Labeling: Apply connected component labeling on the point cloud to group points that are spatially close and similar to one another. This step is done using a breadth-first search or depth-first search algorithm, which starts from an arbitrary point in the point cloud and visits all of its neighboring points that belong to the same cluster.

- Labeling: Assign a unique label to each cluster obtained from the connected component labeling.

- Scale Reduction: Check if the current scale is equal to the minimum scale. If not, reduce the scale parameter by a predefined factor and go back to step 2.

- Output: If the current scale is equal to the minimum scale, the MSCC algorithm is done, and the output is a set of clusters, each of which represents a connected component of points at a particular scale.

In summary, MSCC is a powerful method for segmenting point clouds by iteratively applying a connected component labeling algorithm at multiple scales, it can handle noise and partial occlusions, and it allows for capturing multi-scale features. However, the choice of scale parameter is crucial and should be chosen based on the resolution and density of the point cloud, as well as the size and shape of the objects of interest.

Additionally, the MSCC algorithm has several advantages.

§ 3.5.3 Advantages of MSCC Algorithm

The MSCC algorithm provides a number of benefits, some of which are as follows:

- MSCC is a multi-scale method, it may divide a point cloud into sections of various resolutions. As a result, it can handle a variety of point cloud datasets and adjust to varying granularities of the data.

- By utilizing the voxelization step and linked component labeling technique, MSCC can segment enormous point cloud datasets in a reasonable amount of time, making it appropriate for real-time applications.

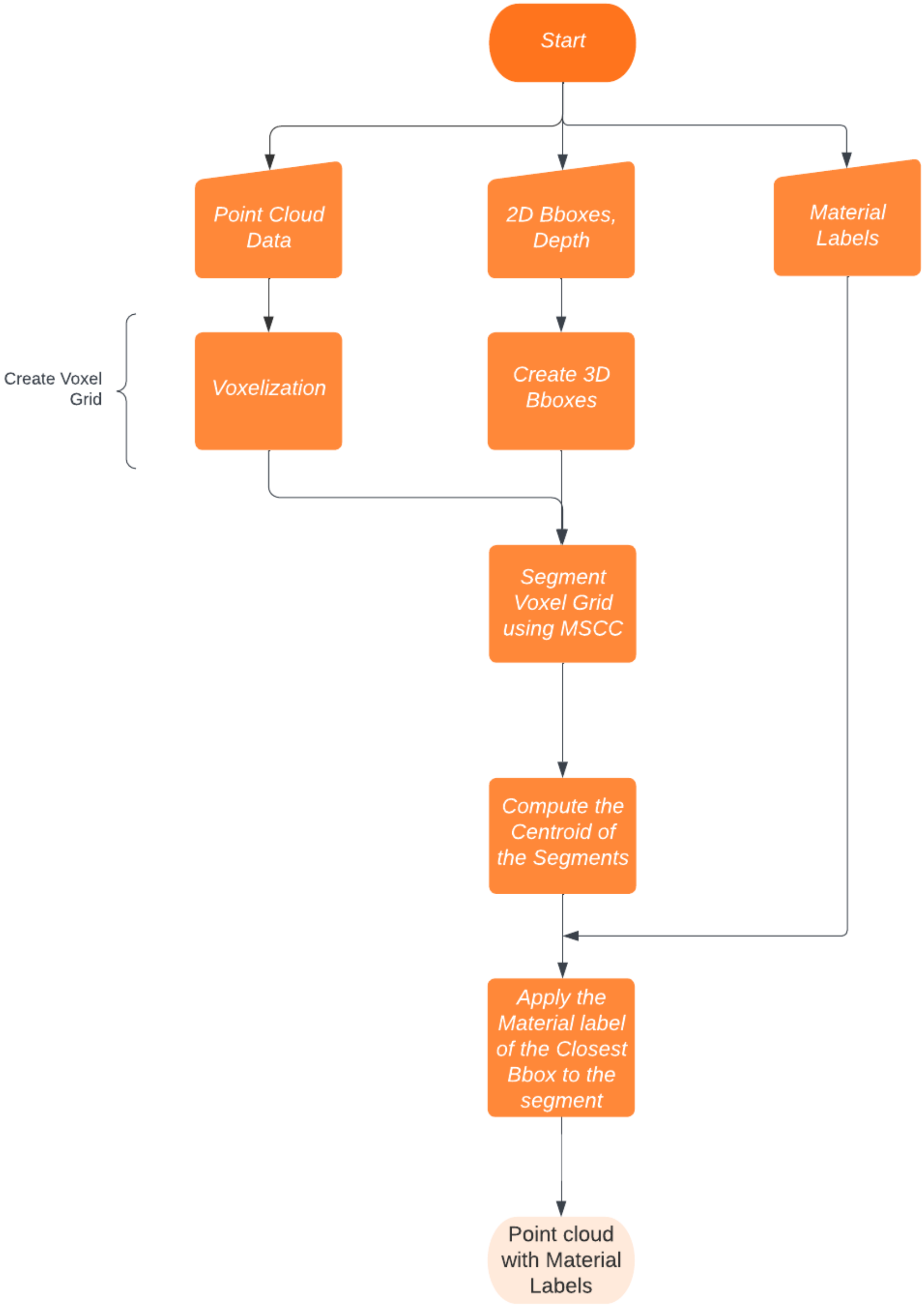

Start
Point Cloud Data
2D Bboxes, Depth
Material Labels
Create Voxel Grid
Voxelization
Create 3D Bboxes
Segment Voxel Grid using MSCC
Compute the Centroid of the Segments
Apply the Material label of the Closest Bbox to the segment
Point cloud with Material Labels

Figure 3.6: Flowchart of Voxel-Based Matching Component

- MSCC doesn't require as many user-defined parameters as some other algorithms. As a result, it is simpler to use and less prone to mistakes than other algorithms that call for a lot of human parameter setting.

- MSCC can handle unorganized data and preserve regions of interest by preserving the structure of the input data. It is also resistant to noise, outliers, and various densities.

- MSCC is a general algorithm that can be applied to a wide range of point clouds and is not restricted to any one kind. It is also independent of the point cloud data-generating sensor.

- Object recognition, scene understanding, and change detection in 3D point cloud data are just a few of the many uses for MSCC.

After matching the 3D coordinates of the point cloud with the 3D bounding boxes obtained from the object detection using the voxel-based matching component, the MSCC algorithm is applied to the voxel grid. The algorithm is applied at multiple scales, starting from a large scale and gradually reducing it to capture more fine-grained details in the point cloud. At each scale, the algorithm applies connected component labeling to group points that are spatially close and similar to one another. These groups of points are then assigned a unique label, and the scale is reduced until the minimum scale is reached.

In the next step, the material labels obtained from object detection (Yolov5) are propagated to each cluster obtained from the MSCC algorithm. This is done by finding the closest bounding box for each cluster and assigning the material label of that bounding box to the cluster. This step is important as it enables the creation of a point cloud map with material labels, which can be used for various applications such as robot navigation, object recognition, and scene understanding.

The process of material label propagation is broken down into the following steps:

- For each segment in the point cloud, compute the centroid of the segment.

- For each 3D bounding box obtained from object detection, compute the centroid of the bounding box.

- For each segment, find the closest bounding box centroid.

- Assign the material label of the closest bounding box to the segment.

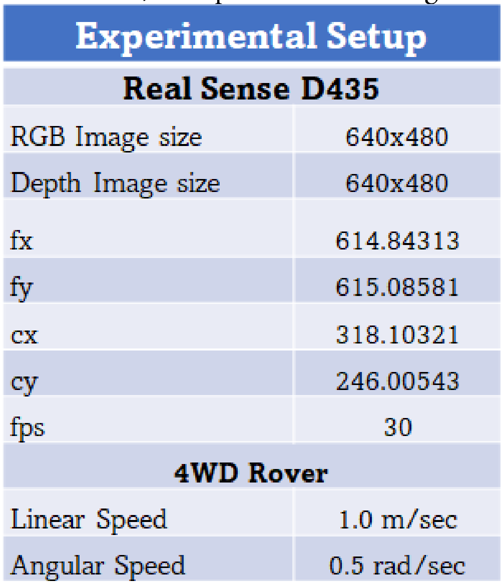
This process can be repeated for all segments in the point cloud, resulting in the accurate propagation of material labels to the corresponding segments in the point cloud. The output of the voxel-based matching component is a point cloud map with material labels, which can be used for various applications. The MSCC algorithm can be used in the pipeline for point cloud segmentation and thus the material labels can be propagated on the point cloud.

CHAPTER 4

EXPERIMENTS AND RESULTS

The research involved conducting experiments on both standard publicly available RGB-D datasets and custom lab RGB-D datasets. The custom lab RGB-D experiments were carried out using a mobile robot navigating indoor scenes, while the standard RGB-D based experiments utilized previously acquired indoor dataset sequences. In addition to these results, the study also includes qualitative results obtained from a publicly available RGB-D dataset. The experiment starts with comprehensive details of the experimental configuration, which was considered crucial for the success of the experiments.

Table 4.1: Experimental Settings



Experimental Setup	
Real Sense D435	
RGB Image size	640x480
Depth Image size	640x480
fx	614.84313
fy	615.08581
cx	318.10321
cy	246.00543
fps	30
4WD Rover	
Linear Speed	1.0 m/sec
Angular Speed	0.5 rad/sec

A list of these settings is shown in Table 4.1. After the parameters were established, the study presented the experimental setup, various datasets used at each component level, and then presented extended mapping results, which were the main focus of the research. The Rover Robotics Platform used for dataset creation is shown in Figure 4.1.

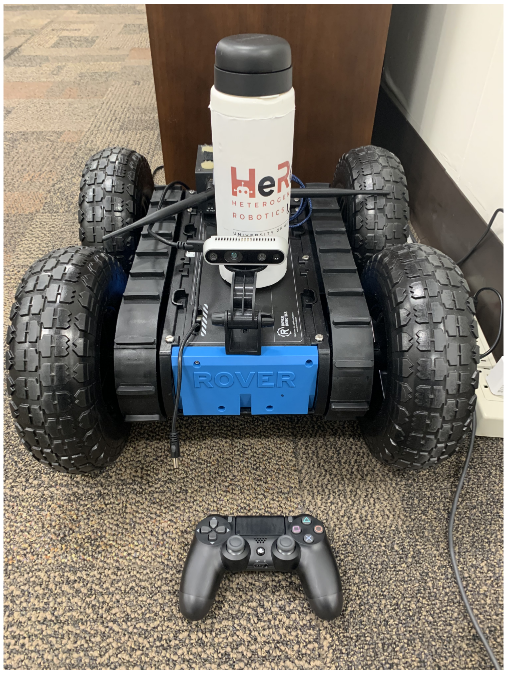


Figure 4.1: Rover Robotics Platform

§ 4.1 Experimental setup

Our method employed a robotic platform with a 4WD Rover base, equipped with a D435 RGB-D camera and LiDAR sensors. The experiments were conducted integrated with ORB SLAM2 framework for the map generation ran on a laptop with Ubuntu 18.04, Intel Core i7, and Nvidia GeForce GTX 1050 Ti. The goal was to enhance maps with object material information that doesn't change over time, such as doors, desks, chairs, and a few other objects.

§ 4.2 Datasets

§ 4.2.1 Object Detection dataset and Yolov5 Pre-trained Model

In this section, we present the object detection module, the Yolov5 model, and the dataset on which the model is trained.

For object detection, we have used the Yolov5 model pre-trained on the standard COCO dataset. The Common Objects in Context (COCO) dataset is a large-scale object recognition, segmentation, and captioning dataset. It was created to support research in computer vision and related fields, and has been widely used for training and evaluating object detection models. The COCO dataset contains over 330K images with 2.5 million object instances, annotated with 80 object categories, including everyday objects such as cars, animals, furniture, and sports equipment. The annotations in the COCO dataset also include image-level captions, object segmentations, and keypoint annotations, which provide rich information for various computer vision tasks.

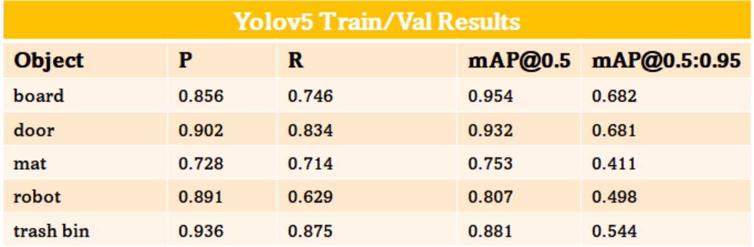
Some sample Yolov5 object detection results are shown in Figure 4.2.

The large scale, high resolution, and diverse categories of the COCO dataset are used to train a YOLOv5 model to detect a wide range of objects with high accuracy. Using this dataset to train a YOLOv5 model helps the model's performance on real-world object detection tasks, such as self-driving cars, surveillance systems, and robotics. Furthermore, by using a high-quality and diverse dataset like COCO, the model may learn to generalize better, so it performs well on unseen data as well.

Semantic indoor sequences can often be of the office and household objects in indoor environments. However, accurately mapping and locating these objects can be difficult due to occlusions and large distances between objects. To overcome these challenges, our method employs the YOLOv5 network Jocher et al., 2022 for semantic object detection in every frame. The network has been trained on the COCO dataset Lin et al., 2014 and can distinguish between 80 different classes of objects.

We created a custom YOLOv5 model based on the pre-trained YOLOv5s model on the COCO dataset and with additional classes, such as board, door, mat, robot, and trash bin. We collected a set of 500 images from the custom dataset and labeled the objects in these images using a LabelImg annotation tool. We augmented these data using rotating and flip techniques and obtained a dataset of 1500 images. The training results for the additional classes are shown in Table 4.2 and the comparison of yolov5 detections with the pre-trained COCO model and the model fine-tuned with additional classes on conference room data sequence is shown in Table 4.3.

Table 4.2: Yolov5 Training Results - Additional Classes



Yolov5 Train/Val Results				
Object	**P**	**R**	**mAP@0.5**	**mAP@0.5:0.95**
board	0.856	0.746	0.954	0.682
door	0.902	0.834	0.932	0.681
mat	0.728	0.714	0.753	0.411
robot	0.891	0.629	0.807	0.498
trash bin	0.936	0.875	0.881	0.544

Table 4.3: Yolov5 - Model Comparison of Object Detections in Conference Room

Yolov5 Object Detections		
Object	**Pre-trained COCO Model**	**Model with Additional Classes**
cloth sheet	40	52
cardboard boxes	128	203
chair	132	118
door	0	72
desks	247	337
mat	0	155
plastic board	1	123
screen	65	107
posters	72	145
robots	0	65

The bounding boxes of the detected objects from the RGB images are used to extract the corresponding cropped images of the depth images. Some examples of cropped RGB and Depth images are shown in Figure 4.3. Normalized depth images are shown in Figure 4.3 for visualization.

The cropped RGB and Depth images are used as the input to the Material Classification Network and the material labels corresponding material types are returned as the output of the network.

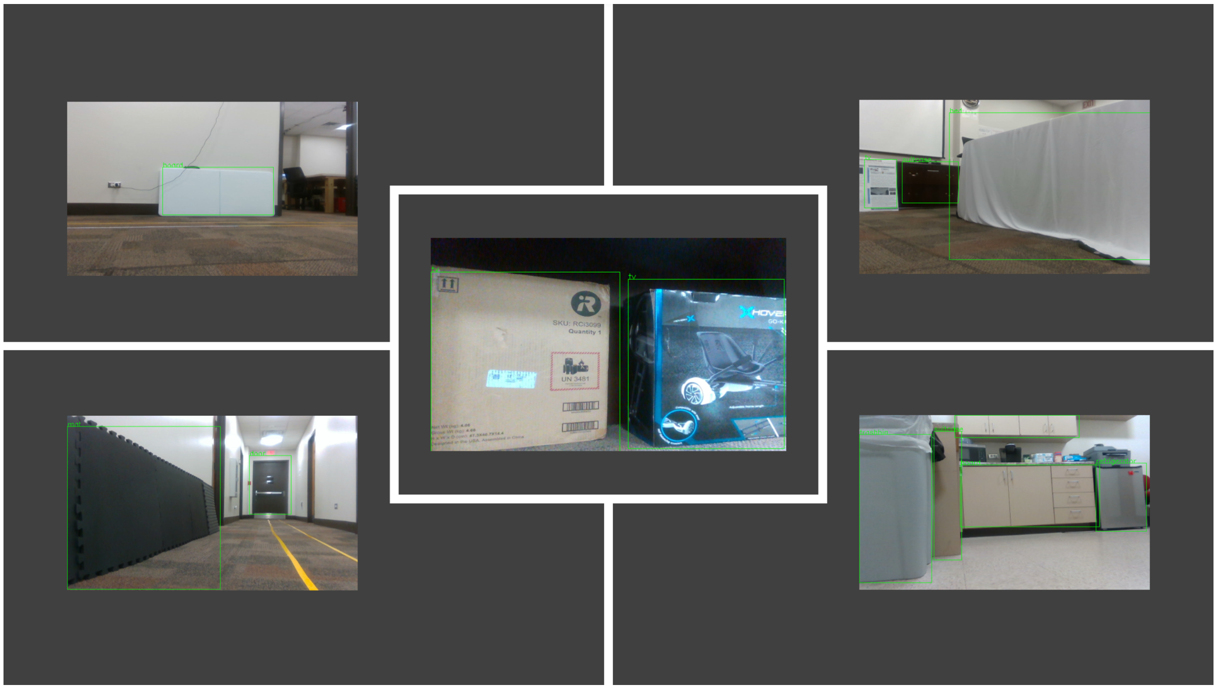


Figure 4.2: Example Yolo Detections of the Custom RGB-D Dataset

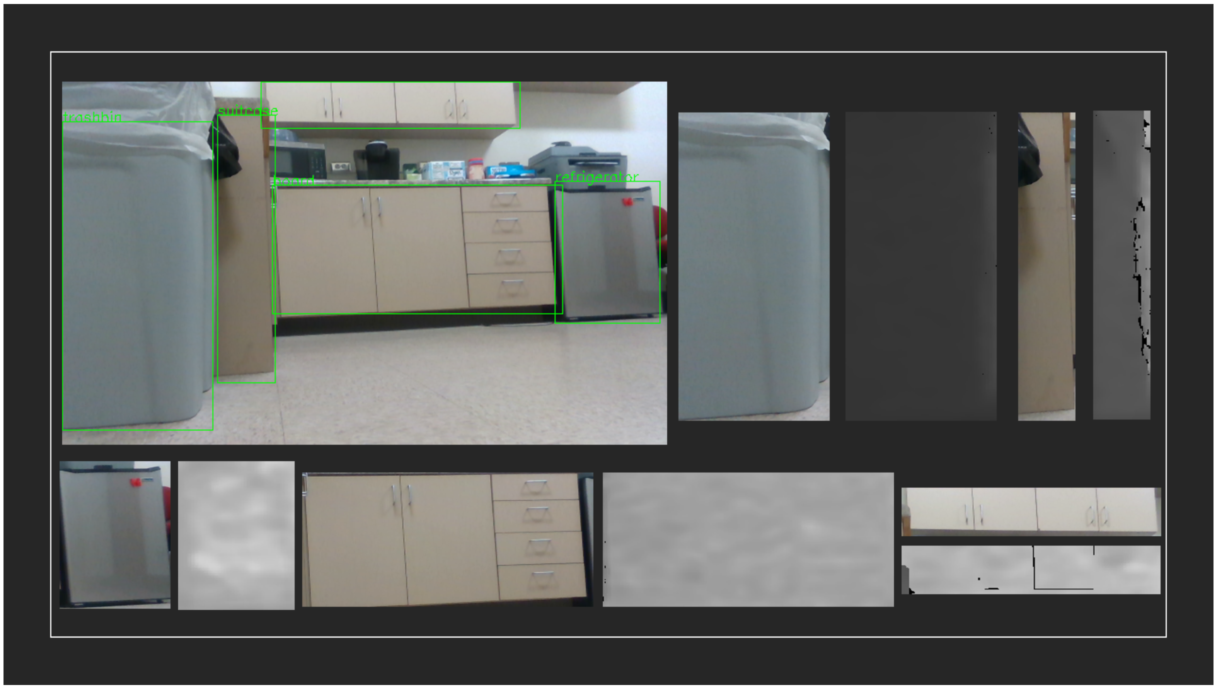


Figure 4.3: Example Yolo Detection and its corresponding cropped RGB and Depth Images

§ 4.2.2 RGB-D Object dataset

We performed our material classification experiments on the Washington RGB-D dataset Lai et al., 2011 captured by Microsoft Kinect. The dataset consists of 300 household objects, grouped into 51 categories. Each object is imaged from 3 vertical angles as well as multiple horizontal angles, resulting in roughly 600 images per object.

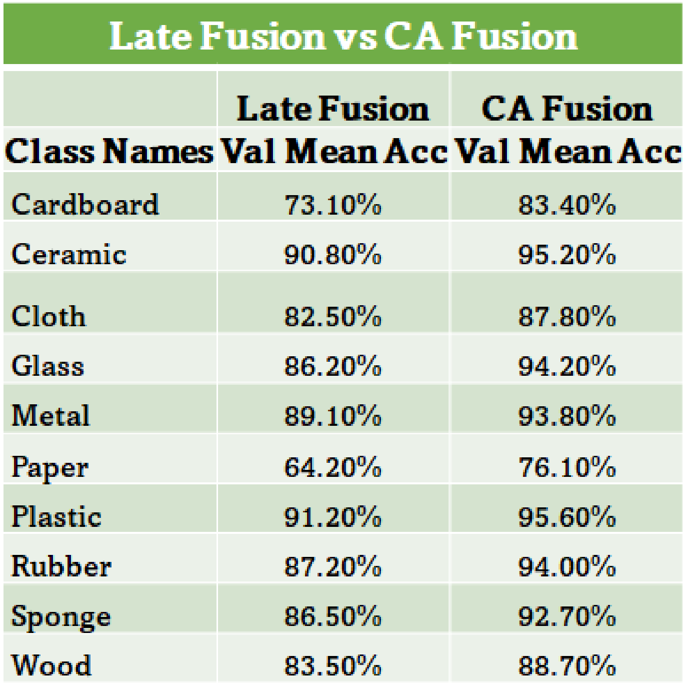
However, in our experiments, we grouped the objects together based on their material type. The dataset is categorized into 10 material types. They are categorized as follows: Cardboard, Ceramic, Cloth, Glass, Metal, Paper, Plastic, Rubber, Sponge, Wood. For Wood class, we extracted the RGB and Depth images from the RGB-D Scenes V2 dataset. We created a balanced dataset with each class consisting of 2500 RGB images and 2500 depth images. We evaluate our method on the material category recognition task, using five cross-validation splits. Each split consists of roughly 4,250 training images and 750 images for testing. During the test time, the task of the model is to assign the correct class label to a previously unseen object instance. Table 4.3 shows the average accuracy of multi-modal CNN in comparison to the two fusion techniques employed for the RGB and Depth networks.

§ 4.2.3 TUM RGB-D and Custom RGB-D datasets

The TUM RGB-D dataset is a comprehensive collection of 39 real-world indoor sequences, categorized into Handheld SLAM, Robot SLAM, and Dynamic Objects. These sequences were captured using an RGB-D Kinect camera in diverse office environments, providing a robust test bed for a wide range of challenges in image processing and computer vision. The dataset features varying camera motion, and environmental conditions such as illumination changes, and includes issues such as image noise, motion blur, occlusion, and more, making it an ideal resource for examining the performance of algorithms and techniques in realistic scenarios.

To rigorously test the efficacy of our proposed method, we have compiled a dataset of five distinct indoor environments. This dataset was generated through the teleoperation of a robot, capturing raw sensor information from two RGB-D cameras, LiDAR, and odometry, using the rosbag toolkit. The dataset encompasses a diverse range of objects commonly found in indoor spaces, such as chairs, doors,

Table 4.4: Comparison of Late Fusion vs CA Fusion



Late Fusion vs CA Fusion

	Late Fusion	CA Fusion
Class Names	Val Mean Acc	Val Mean Acc
Cardboard	73.10%	83.40%
Ceramic	90.80%	95.20%
Cloth	82.50%	87.80%
Glass	86.20%	94.20%
Metal	89.10%	93.80%
Paper	64.20%	76.10%
Plastic	91.20%	95.60%
Rubber	87.20%	94.00%
Sponge	86.50%	92.70%
Wood	83.50%	88.70%

desks, and trash bins, as illustrated in Figures 4.2 and 4.3. By utilizing multiple sensor inputs and covering a range of indoor settings, our dataset provides a robust evaluation of the proposed method's performance in real-world scenarios.

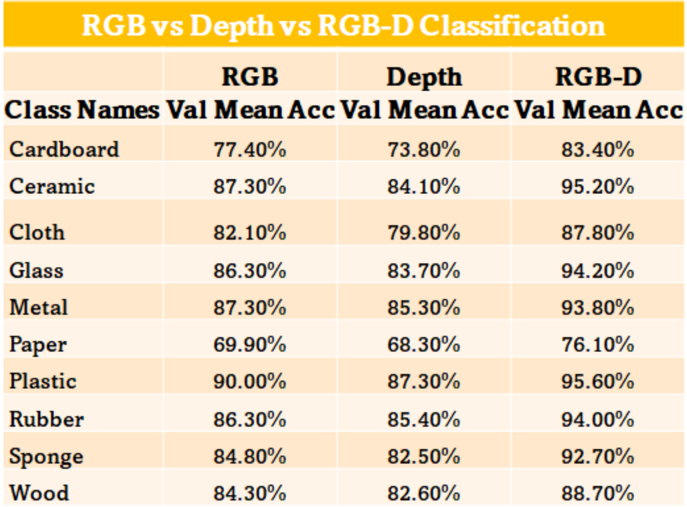
§ 4.3 Material Classification

The RGB and Depth networks consist of five convolutional layers followed by two fully connected layers. Later these networks are fused together using different techniques and finally, a softmax layer is used for classification. Rectified linear units(ReLUs) are used as the activation functions in all but the final classification layer. We implement our experiments using the ResNet50 as the backbone, on a PC with 3 NVIDIA GeForce GPUs. The learning rate, weight decay and mini-batch size are set as 1e-5, 0.0005 and 4 respectively. 350 epochs are used for training, with training time of approximately 16.5 hours.

Firstly, we conducted experiments to train the RGB and Depth networks separately and combine the networks to make joint predictions. This is termed the late fusion technique. From the results shown in Table it can be seen that with late fusion, the model achieves an overall accuracy of 82.6 ± 1.6%.

Later we conducted experiments for material classification using CA Fusion modules. Cascading of multi-level features successively without intermediate level-wise supervision results in ambiguous multi-level combinations. The high-level contexts are not well incorporated into shallow layers. Adding intermediate supervision, the multi-modal fusion network can learn level-specific predictions. Visually, the shallow layers are able to identify edge information and the deep layers can learn global contexts to detect the object accurately. The CA-Fuse module involves cross-modal residual connections and complementarity-aware supervisions, captures the cooperated information, and boosts better cross-modal combination, thus generating more accurate object detection and classification. The material classification results indicate that the model benefits from a more sufficient fusion of multi-modal and multi-level features, and hence CA fusion models achieve much better performance than the late fusion models.

Table 4.5: CA Fusion - Material Classification Training Results

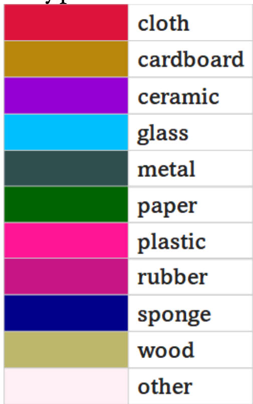


RGB vs Depth vs RGB-D Classification

	RGB	Depth	RGB-D
Class Names	Val Mean Acc	Val Mean Acc	Val Mean Acc
Cardboard	77.40%	73.80%	83.40%
Ceramic	87.30%	84.10%	95.20%
Cloth	82.10%	79.80%	87.80%
Glass	86.30%	83.70%	94.20%
Metal	87.30%	85.30%	93.80%
Paper	69.90%	68.30%	76.10%
Plastic	90.00%	87.30%	95.60%
Rubber	86.30%	85.40%	94.00%
Sponge	84.80%	82.50%	92.70%
Wood	84.30%	82.60%	88.70%

The multi-modal CNN, using the complementarity-aware fusion, (CA Fus-CNN) yields an overall accuracy of 91.5 ± 1.6% when using RGB and depth (83.1 ± 2.6% and 81.8 ± 2.3% when used only the RGB or depth modality respectively), which – to the best of our knowledge – is the highest accuracy reported for this dataset to date. We also report results for combining the RGB and Depth networks with a late fusion (Late Fus-CNN). It can be seen in the table, this technique yielded a decreased performance. Overall our experiments show that the CA Fusion model performs significantly better than the late fusion model for material classification.

Table 4.6: Material types and their color representation



🟥	cloth
🟫	cardboard
🟪	ceramic
🟦	glass
⬛	metal
🟩	paper
🟥	plastic
🟥	rubber
🟦	sponge
🟨	wood
⬜	other

§ 4.4 Material Mapping Results

The experiments are conducted to evaluate the performance of the proposed method for semantic mapping in controlled conditions. To do so, we collected a dataset consisting of six indoor sequences captured while a robot was being teleoperated. Each sequence included raw data from multiple sources including two RGB-D cameras, LiDAR, and odometry.

The sequences contained different types of objects, including doors, desks, boards, mat, trash bins, posters, and a few other objects, as shown in the accompanying images. The static objects (all except people and chairs) had their location specified in a ground truth map provided by the authors.

The Intel Real sense D435 RGB-D camera is used to capture the sequences. The dataset was created as the majority of existing datasets for semantic mapping and 3D object detection did not include doors and other objects of interest for navigation in indoor environments.

The following are some qualitative results of our proposed method, based on sequences from the datasets we introduced.

RGB-D TUM:

Multiple objects are successfully detected and represented as semantic labels on the map with their corresponding material type. The colors used to represent each material type is shown in Table 4.4. The material type predictions with max probability less than 0.5 are considered as other category class. We were also able to accurately estimate the size of objects by their point clouds, which gives an idea of their general dimensions. For example, the phone and book objects are depicted correctly with their sizes and the material labels are projected onto the point cloud, with the help of the connected components algorithm.

Table 4.7: List of Objects and their Material Type in Conference Room

Conference Room		
Object	**Material Type**	**Count**
cloth sheet	cloth	1
cardboard boxes	cardboard	8
chair	fiber	3
door	wood	2
desks	wood	3
mat	rubber	1
plastic board	plastic	1
screen	polyester	1
posters	paper	3
robots	metal	2

The objects available in the Conference room, their actual material type, and the number of objects in each class are listed in Table 4.7.

Table 4.8: Conference Room - Object Detections and Material Classification

Conference Room						
Object	**Detections**	**Material Prediction**	**IoU**	**mAP**	**TP**	**FP**
cloth sheet	52	cloth	0.67	0.643	33	19
cardboard boxes	203	cardboard	0.784	0.622	126	77
chair	118	other	0.823	0.613	72	46
door	72	wood	0.841	0.648	47	25
desks	337	wood	0.82	0.673	226	111
mat	155	rubber	0.776	0.631	98	57
plastic board	123	plastic	0.872	0.646	79	44
screen	107	other	0.864	0.672	72	35
posters	145	paper	0.88	0.593	86	59
robots	65	metal	0.732	0.66	43	22

The Yolov5 detections of the objects in the Conference Room and their material predictions are shown in Table 4.8. The average IoU and mAP across various objects are 0.806 and 0.64 respectively. The number of keyframes from the Conference Room dataset sequence used by ORB-SLAM2 for creating point clouds is 323.

Table 4.9: List of Objects and their Material Type in Kitchen Room

Kitchen Room		
Object	**Material Type**	**Count**
bottles	plastic	2
cardboard boxes	cardboard	4
coffee machine	metal	1
cupboard	wood	3
door	wood	2
desks	wood	1
microwave	metal	1
printer	metal	1
plastic board	plastic	1
refrigerator	metal	1
trash bin	plastic	2
posters	paper	3

The objects available in the Kitchen room, their actual material type, and the number of objects in each class are listed in Table 4.9.

Table 4.10: Kitchen Room - Object Detections and Material Classification

Kitchen Room						
Object	**Detections**	**Material Prediction**	**IoU**	**mAP**	**TP**	**FP**
cardboard boxes	120	cardboard	0.674	0.652	78	42
cupboard	220	wood	0.838	0.646	142	78
door	52	wood	0.824	0.692	36	16
desks	43	wood	0.856	0.66	29	14
mat	4	rubber	0.74	0.75	3	1
microwave	42	metal	0.82	0.644	27	15
printer	8	metal	0.746	0.622	5	3
plastic board	30	plastic	0.822	0.632	19	11
refrigerator	26	metal	0.8	0.572	15	11
trash bin	24	plastic	0.876	0.658	16	8
posters	62	paper	0.852	0.632	39	23

The Yolov5 detections of the objects in the Kitchen room and their material predictions are shown in Table 4.10. The average IoU and mAP across various objects are 0.804 and 0.651 respectively. The number of keyframes from the Kitchen Room dataset sequence used by ORB-SLAM2 for creating point clouds is 367. It is observed that the Yolov5 model is not able to detect objects like the coffee machine and bottles available in the Kitchen Room.

Table 4.11: List of Objects and their Material Type in Office Room

Office Room		
Object	**Material Type**	**Count**
cardboard boxes	cardboard	5
chair	metal	3
door	wood	2
desks	wood	5
monitor	glass	1
robot	metal	6
posters	paper	2

The objects available in the Kitchen room, their actual material type, and the number of objects in each class are listed in Table 4.11.

Table 4.12: Office Room - Object Detections and Material Classification

Office Room						
Object	**Detections**	**Material Prediction**	**IoU**	**mAP**	**TP**	**FP**
cardboard boxes	118	cardboard	0.782	0.626	74	44
chair	26	metal	0.867	0.532	14	12
door	128	wood	0.723	0.672	86	42
desks	134	wood	0.833	0.658	88	46
monitor	43	glass	0.672	0.654	28	15
posters	28	paper	0.84	0.72	20	8
robot	14	metal	0.765	0.64	9	5

The Yolov5 detections of the objects in the Office Room and their material predictions are shown in Table 4.12. The average IoU and mAP across various objects are 0.783 and 0.643 respectively. The number of keyframes from the Office Room dataset sequence used by ORB-SLAM2 for creating point clouds is 442.

Table 4.13: List of Objects and their Material Type in Experiments Station Room

Experiments Station Room		
Object	**Material Type**	**Count**
cardboard boxes	cardboard	4
cloth sheet	cloth	1
chair	fiber	4
door	wood	3
desks	wood	4
keyboard	plastic	1
mat	rubber	1
metal desk	metal	1
metal plate	metal	1
monitor	glass	2
plastic board	plastic	1
white board	wood	1
posters	paper	1
robot	metal	1

The objects available in the Kitchen room, their actual material type, and the number of objects in each class are listed in Table 4.13.

Table 4.14: Experiments Station Room - Object Detections and Material Classification

Experiments Station Room						
Object	**Detections**	**Material Prediction**	**IoU**	**mAP**	**TP**	**FP**
cardboard boxes	152	cardboard	0.768	0.61	93	59
cloth sheet	24	cloth	0.712	0.666	16	8
chair	206	other	0.816	0.586	121	85
door	154	wood	0.853	0.626	97	57
desks	208	wood	0.81	0.674	140	68
keyboard	98	plastic	0.832	0.666	65	33
mat	156	rubber	0.768	0.638	99	57
metal desk	72	metal	0.845	0.62	45	27
metal plate	88	metal	0.887	0.615	54	34
monitor	29	glass	0.715	0.623	18	11
plastic board	70	plastic	0.834	0.652	46	24
white board	68	other	0.857	0.533	36	32
posters	20	paper	0.813	0.668	13	7
robot	57	metal	0.817	0.657	38	19

The Yolov5 detections of the objects in the Experiments Station room and their material predictions are shown in Table 4.14. The average IoU and mAP across various objects are 0.809 and 0.631 respectively. The number of keyframes from the Experiments Station Room dataset sequence used by ORB-SLAM2 for creating point clouds is 318. In this sequence, our model failed to predict the whiteboard material correctly. The model predicted it as other category.

Table 4.15: List of Objects and their Material Type in Swarm Robots Room

Swarm Robots Room		
Object	**Material Type**	**Count**
cardboard box	cardboard	1
chair	fiber	3
cupboard	metal	1
door	wood	2
desks	wood	2
monitor	glass	1
metal desk	metal	1
cpu	metal	1
white board	wood	1
poster	paper	2

The objects available in the Kitchen room, their actual material type, and the number of objects in each class are listed in Table 4.15.

Table 4.16: Swarm Robots Room - Object Detections and Material Classification

Swarm Robots Room						
Object	**Detections**	**Material Prediction**	**IoU**	**mAP**	**TP**	**FP**
cardboard boxes	72	cardboard	0.863	0.678	49	23
chair	86	other	0.786	0.622	54	32
cupboard	192	metal	0.887	0.61	117	75
door	168	wood	0.69	0.684	115	53
desks	84	wood	0.712	0.642	54	30
monitor	20	glass	0.858	0.676	14	6
metal desk	136	metal	0.831	0.678	92	44
cpu	54	metal	0.882	0.56	30	24
white board	163	wood	0.712	0.644	105	58
poster	138	paper	0.788	0.66	92	46

The Yolov5 detections of the objects in the Swarm Robots room and their material predictions are shown in Table 4.16. The average IoU and mAP across various objects are 0.801 and 0.645 respectively. The number of keyframes from the Swarm Robots Room dataset sequence used by ORB-SLAM2 for creating point clouds is 196.

Figures 4.4 and 4.5 show two additional point clouds from TUM sequences and their semantic material maps. These figures demonstrate that our method can distinguish between small objects of the same class, such as books, bottles, mouse, and cups. There is also a teddy bear present in this scenario, shown in the right back. However, we were not able to validate the material semantics on the TUM datasets as the dataset does not provide information on the material type for the objects available in the data sequence.

Additionally, there were no major works that do semantic material labeling of the point cloud. Thus, we compare our work to the closest relevant work which does object detection on the TUM dataset. We compare the results in regards to semantic object detection with the work Hempel and Al-Hamadi, 2022. The corresponding RGB image, the reference paper result, and ours are shown in Figures 4.6 and 4.7

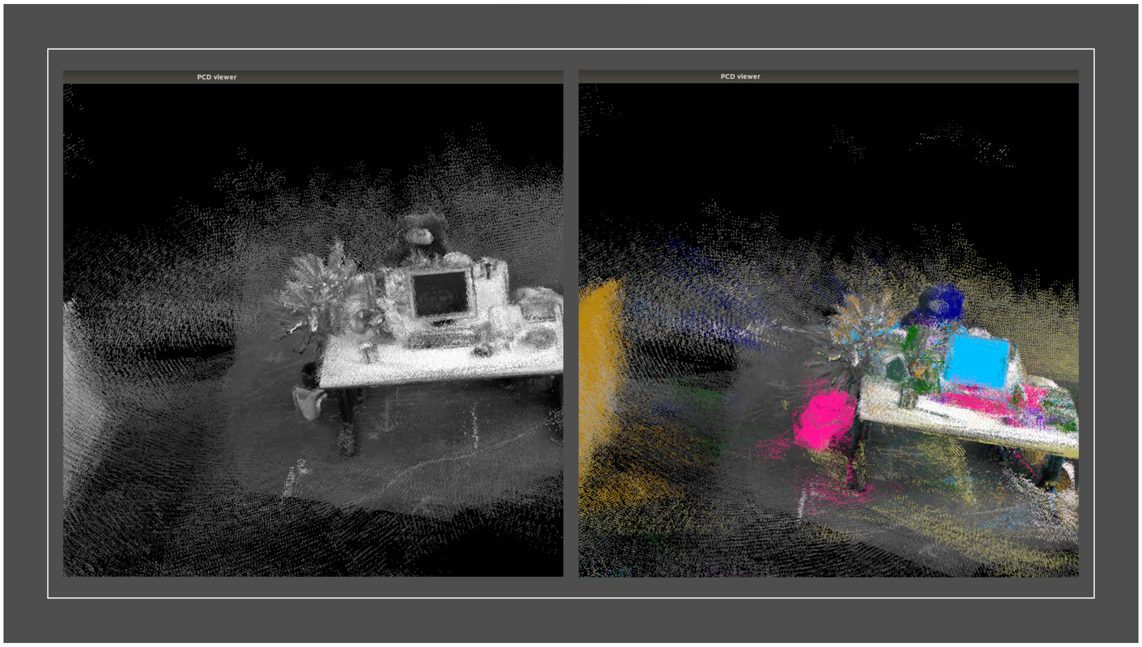


Figure 4.4: Point cloud and semantic material map of fr2_desk sequence

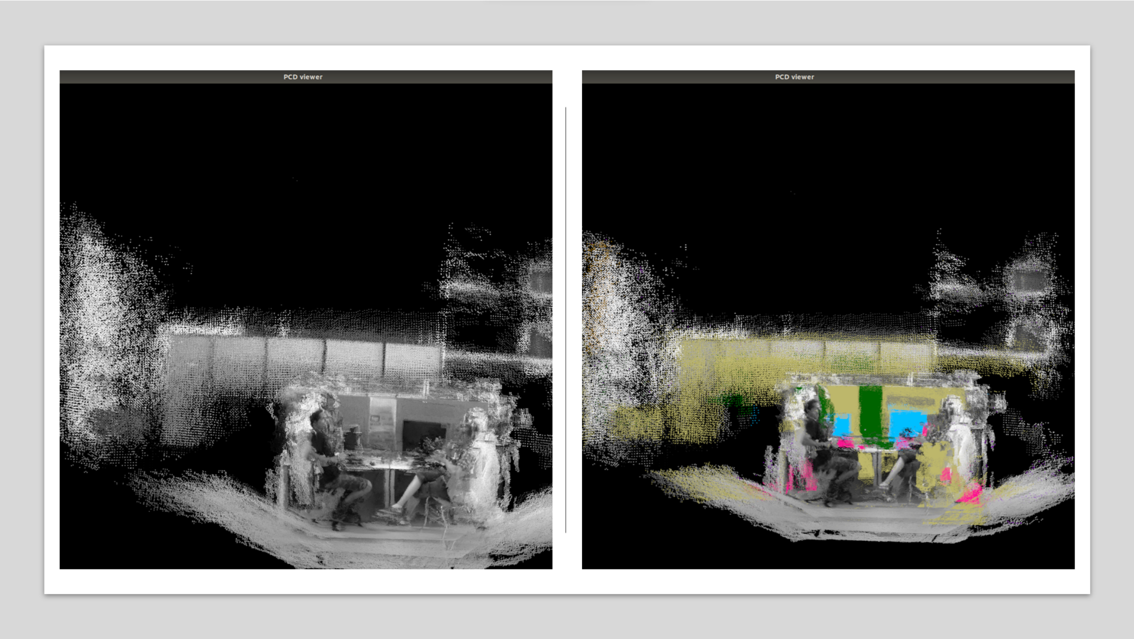


Figure 4.5: Point cloud and semantic material map of fr3_sitting_xyz sequence

respectively. Even though the reference paper detects a few objects, our method detects various objects and also estimates their material type.

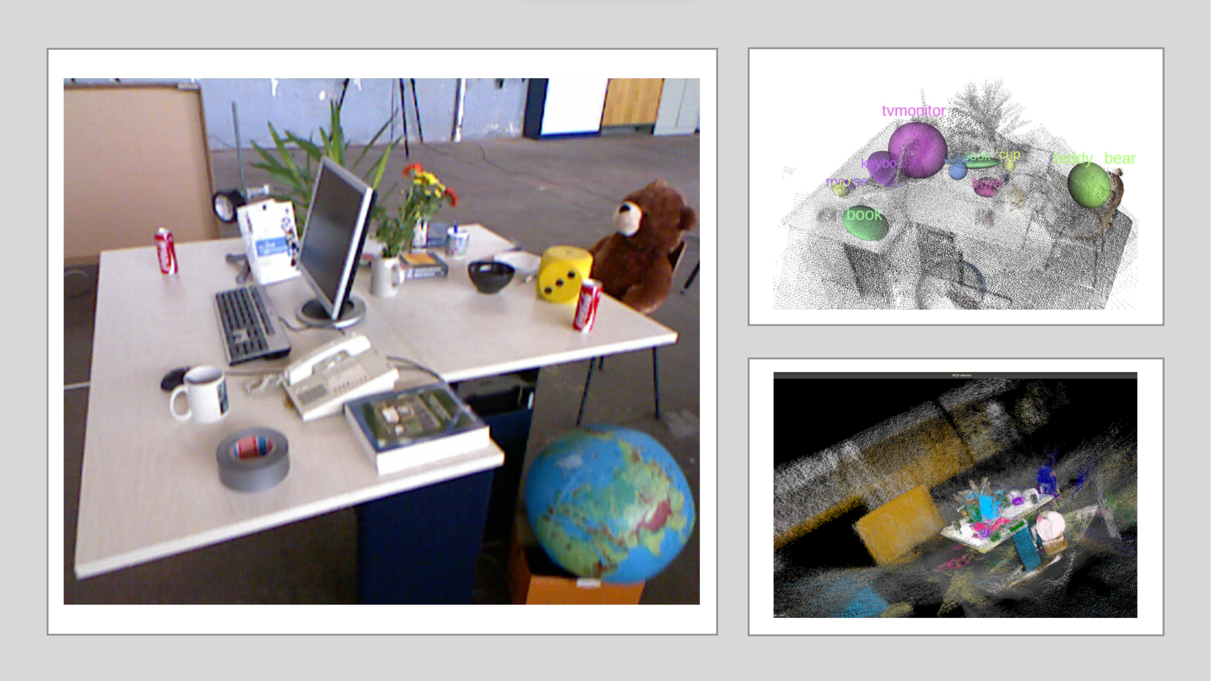


Figure 4.6: RGB Image and Comparison with Hempel and Al-Hamadi, 2022 of fr2_desk sequence

RGB-D Custom

We performed the experiments in five different rooms of our lab environment. Tables 4.8, 4.10, 4.12, 4.14, and 4.16 presents the list of the various objects available in each room, their detections, and material predictions. The mean average precision (mAP) and IoU values are also shown in the tables. Figures 4.8 to 4.17 show the ground truth RGB images labeled, their point cloud, and the results of our method-generated semantic map.

The model was able to classify the material type of different objects. For example, the trash bin in the kitchen and graduate rooms are of different material types, and the chairs in the office room and conference room are of different material types. Also, we can observe that even though the Yolov5 model detections are not accurate in object type, our model was able to classify the material type.

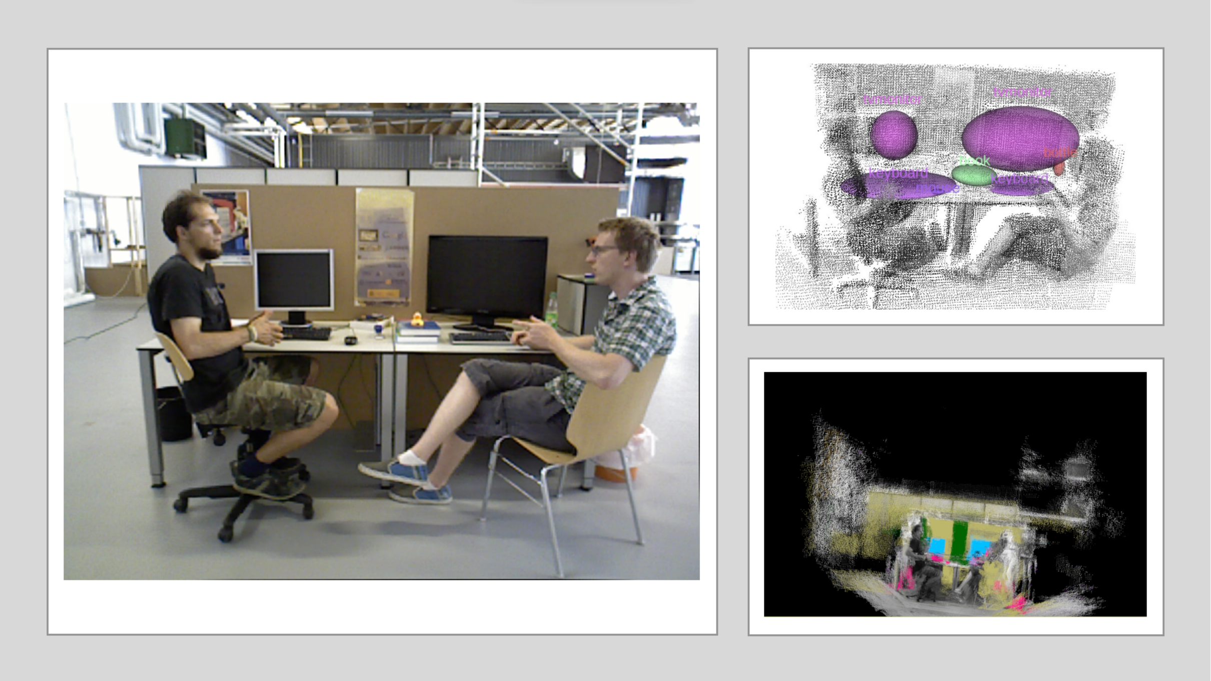


Figure 4.7: RGB Image and Comparison with Hempel and Al-Hamadi, 2022 of fr3_sittting_xyz sequence

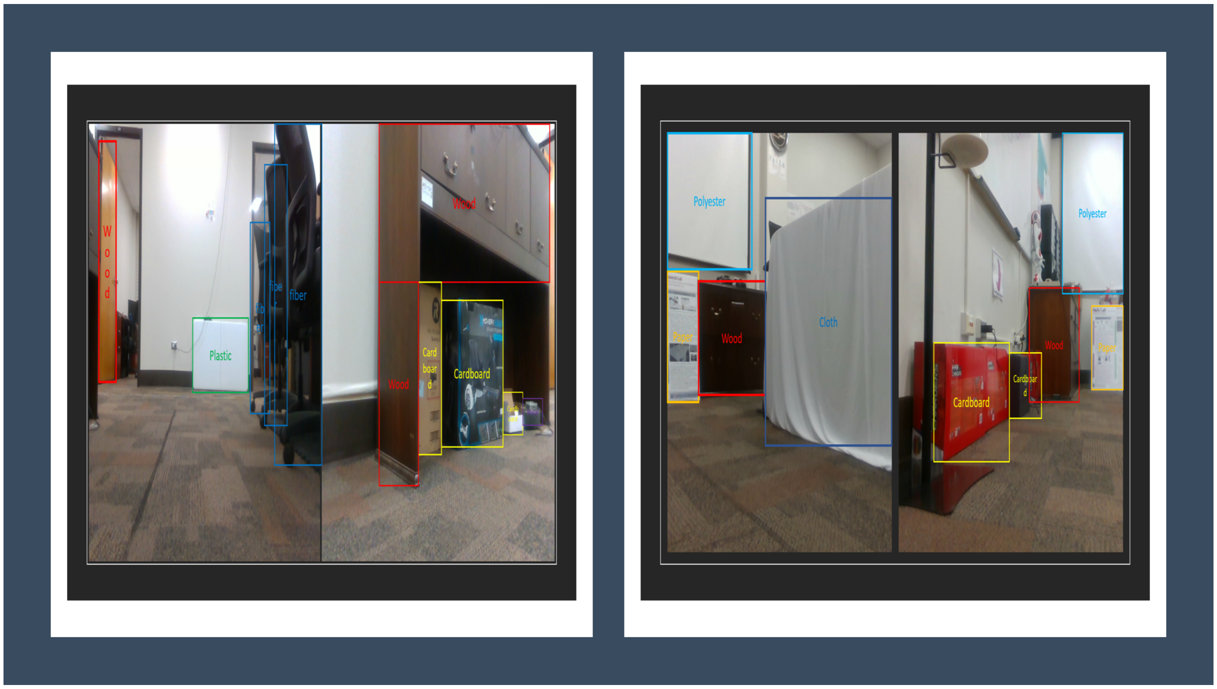


Figure 4.8: RGB Images, Ground Truth of Office Room

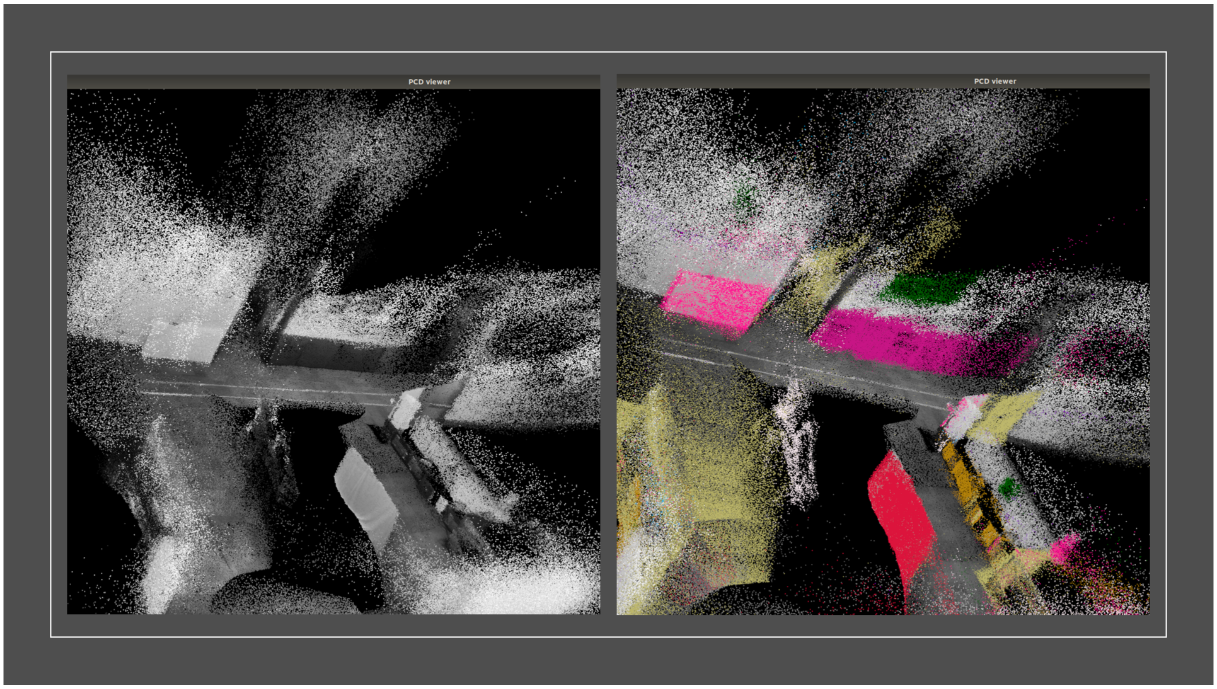


Figure 4.9: Office Room, Point Cloud and Semantic Material Map

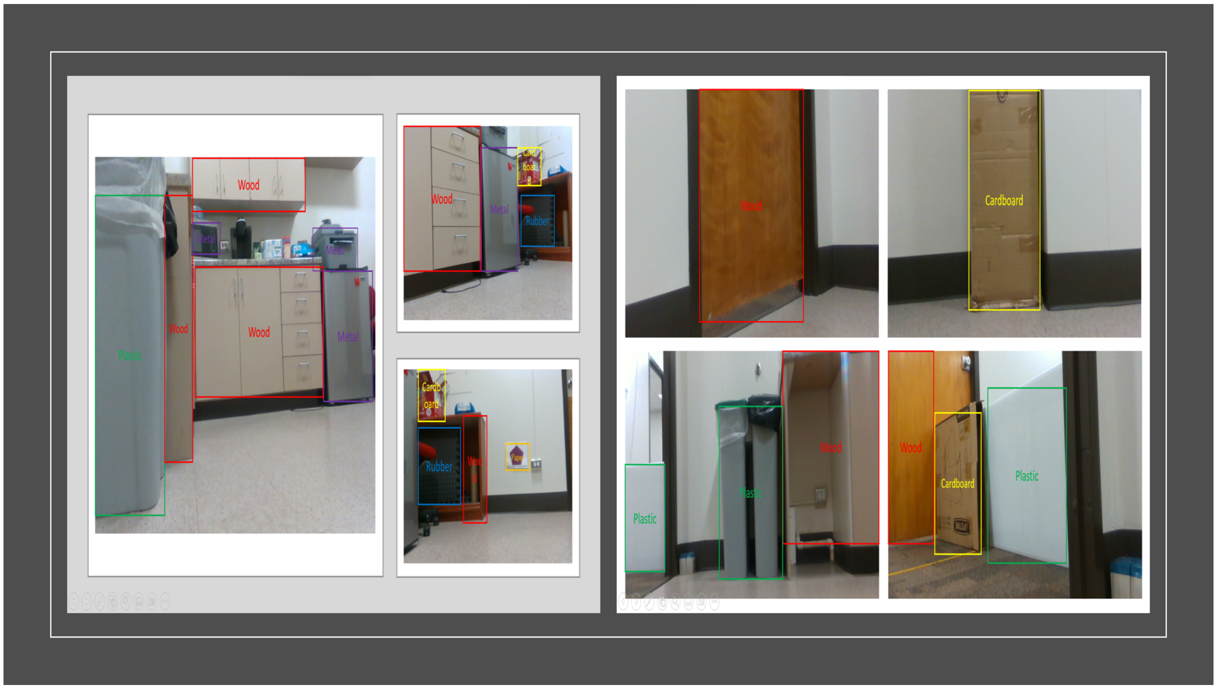


Figure 4.10: RGB Images, Ground Truth of Kit Room

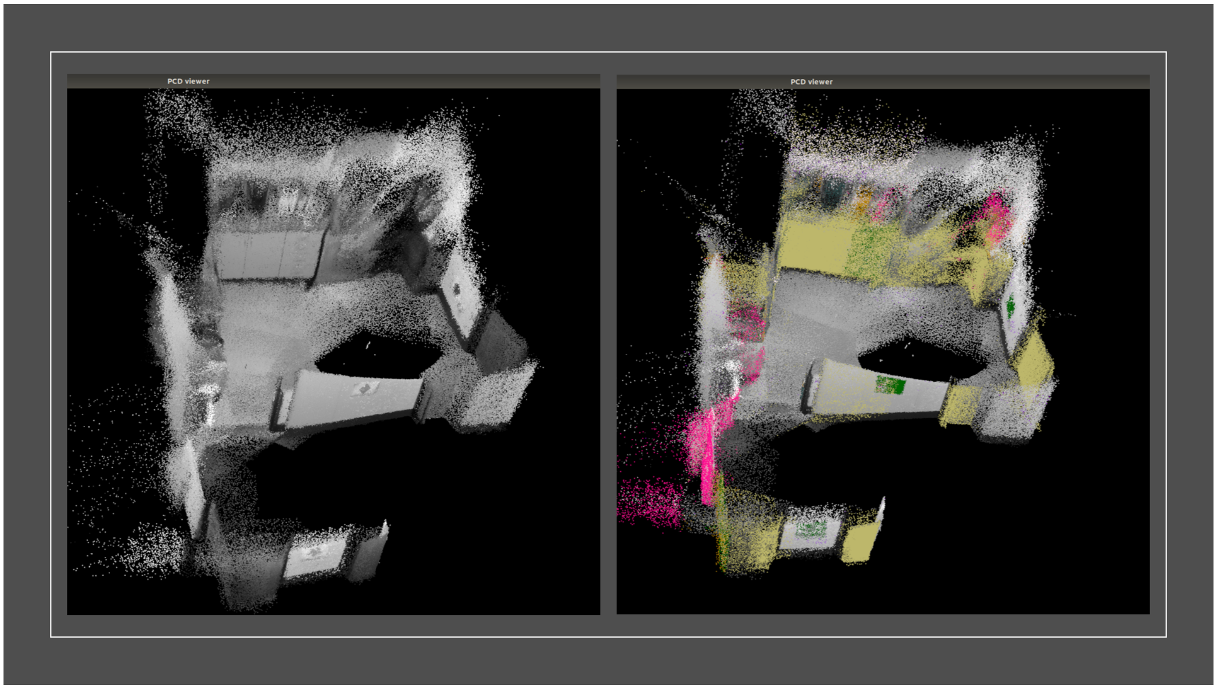


Figure 4.11: Kitchen Room, Point Cloud and Semantic Material Map

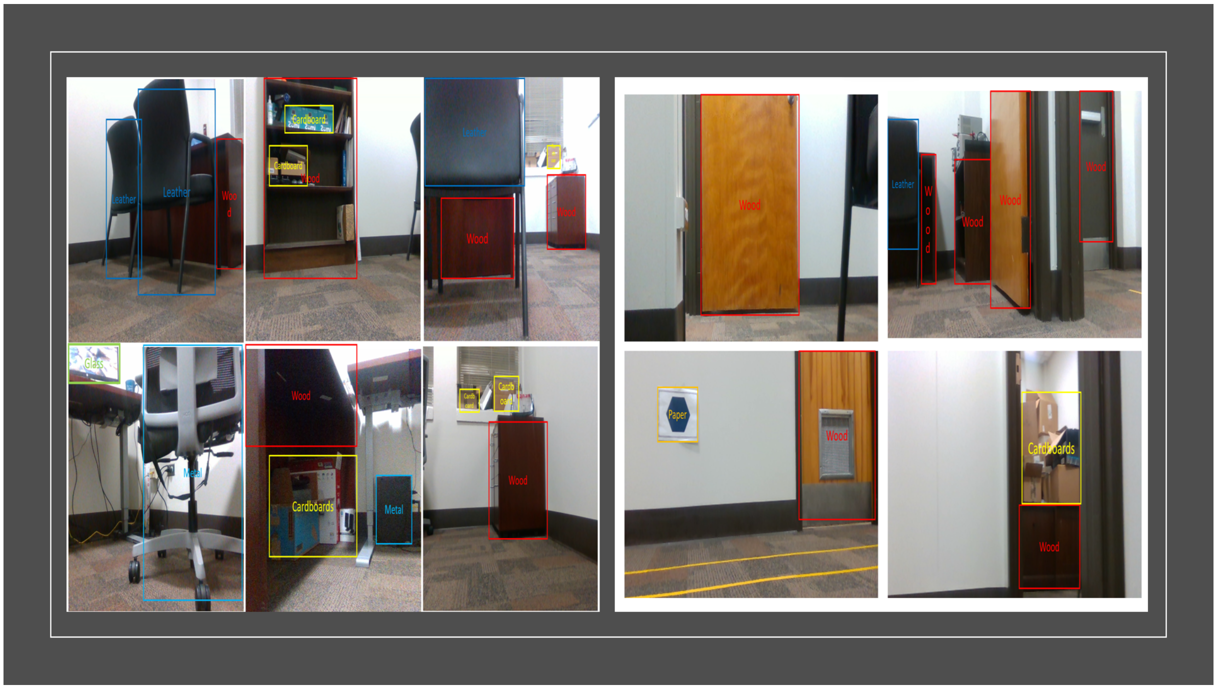


Figure 4.12: RGB Images, Ground Truth of Office Room

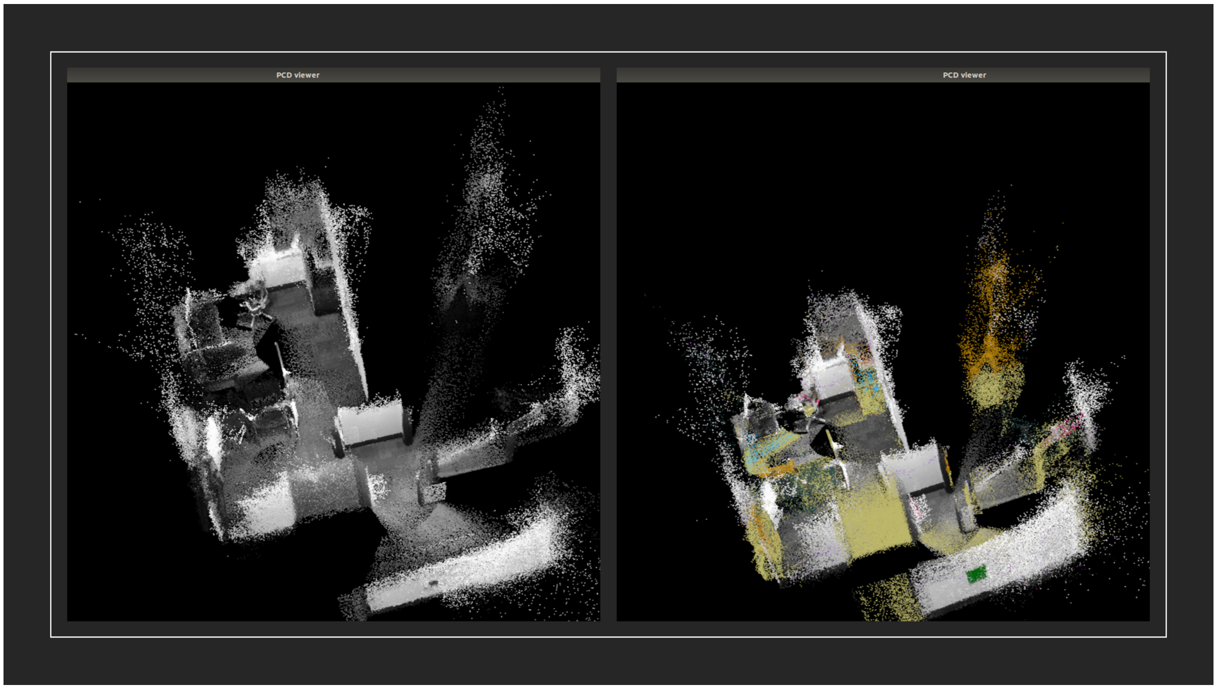


Figure 4.13: Office Room, Point Cloud and Semantic Material Map

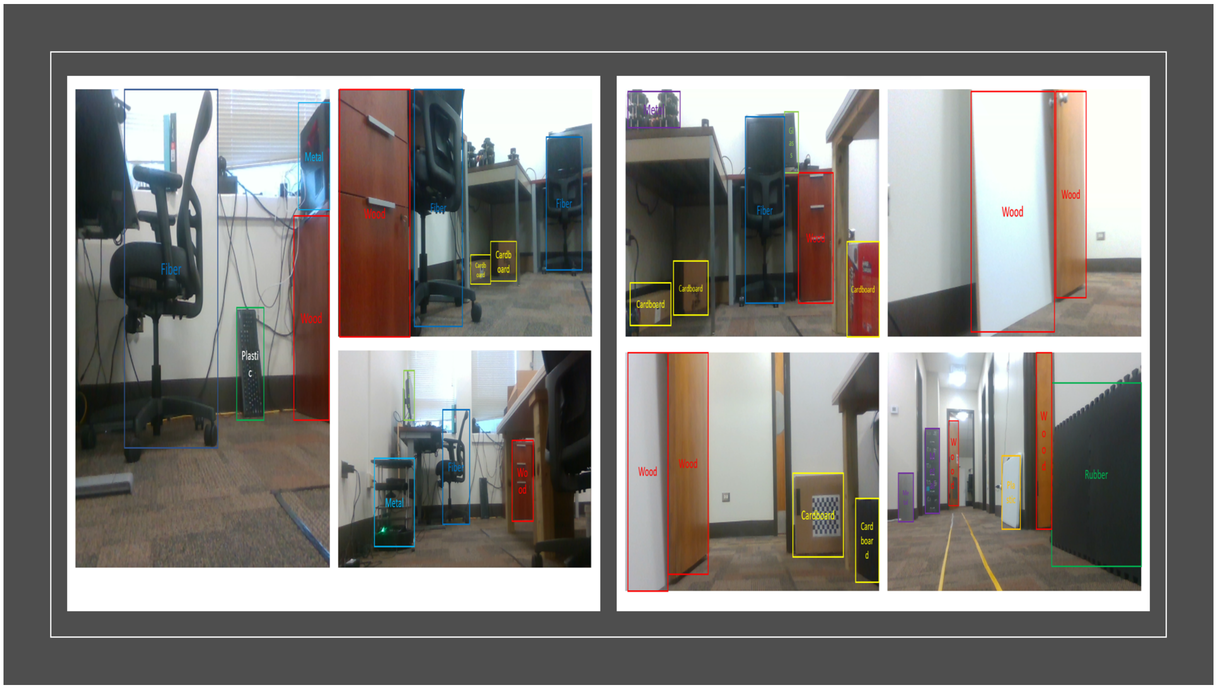


Figure 4.14: RGB Images, Ground Truth of Experiments Station Room

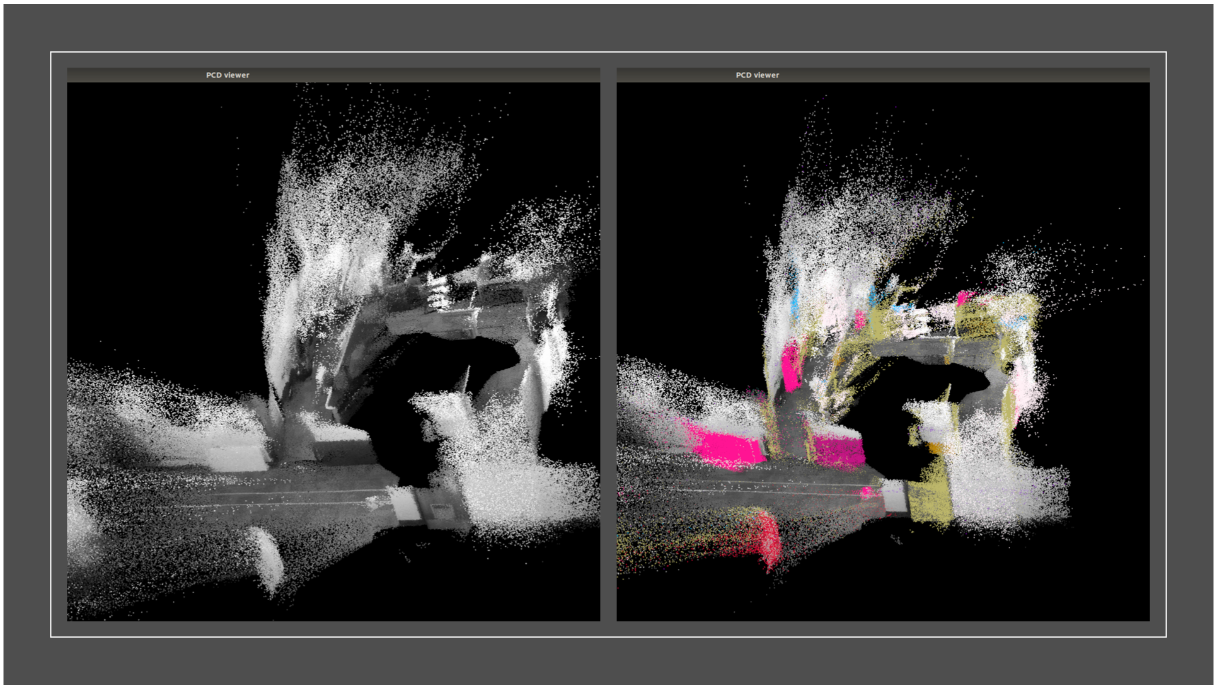


Figure 4.15: Experiments Station Room, Point Cloud and Semantic Material Map

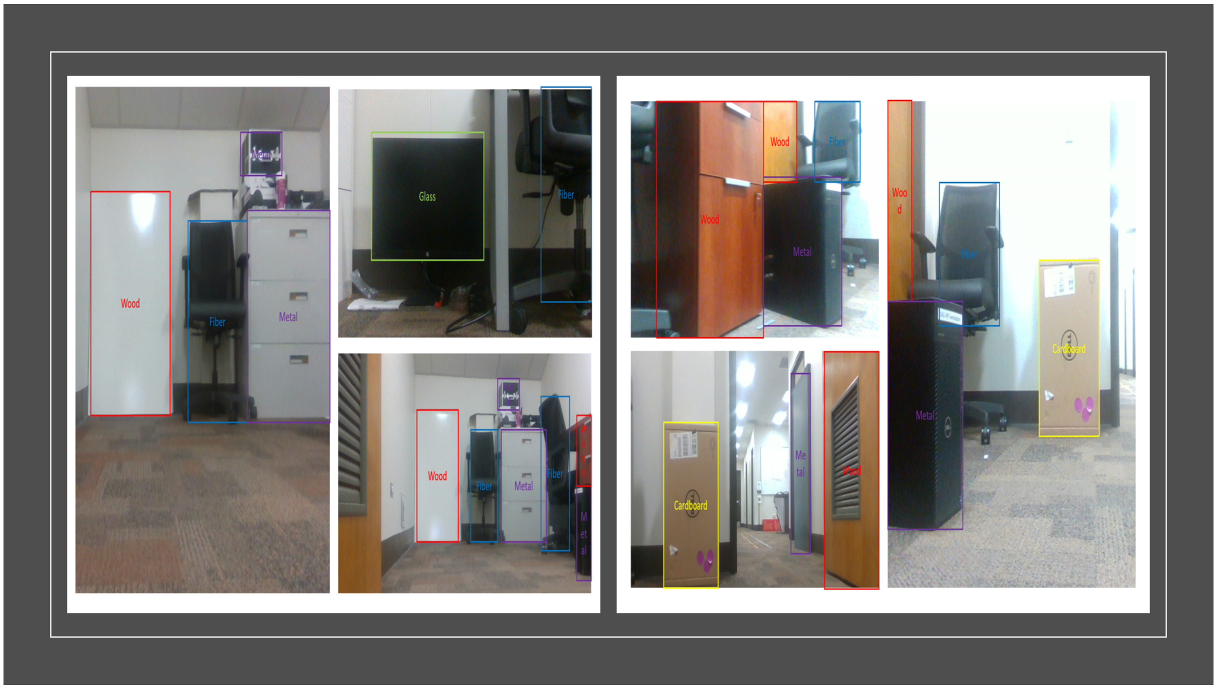


Figure 4.16: RGB Images, Ground Truth of Swarm Robots Room

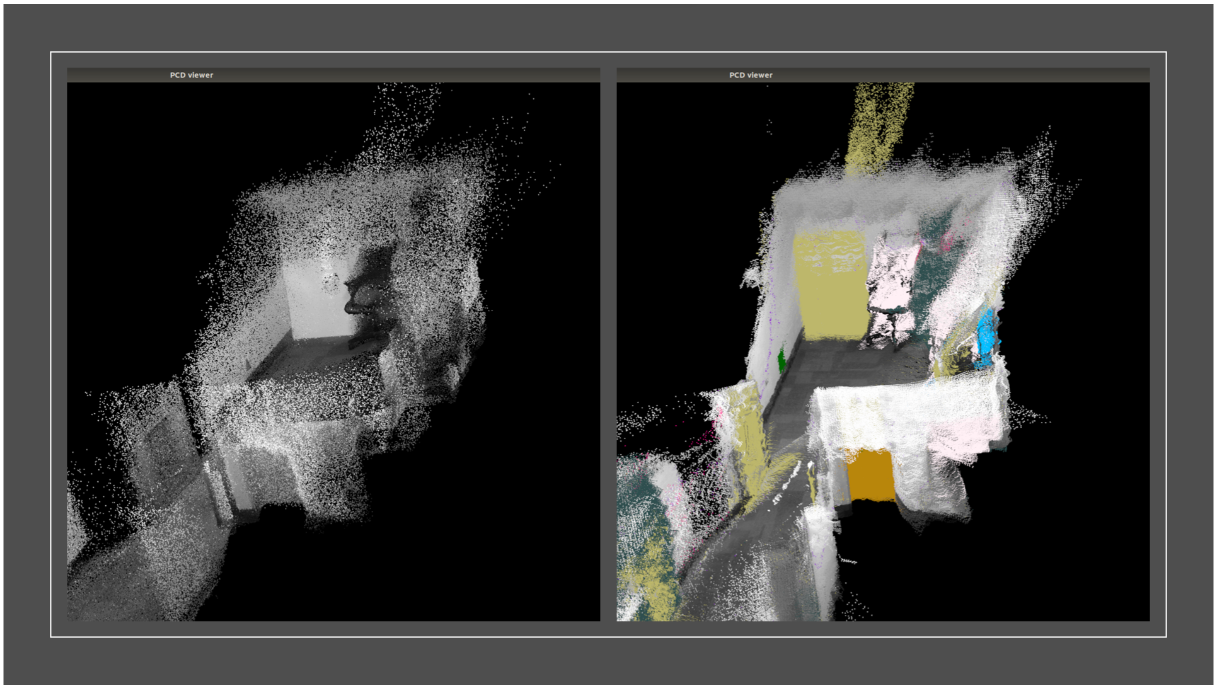


Figure 4.17: Swarm Robots Room, Point Cloud and Semantic Material Map

On the flip side, our model has made some wrong predictions for certain objects of the same material type. For example, the model failed to classify the board object material type as wood in Experiment Station room and was able to able to classify the board object as wood in the Swarm Robots room experiment.

Table 4.17: Comparison of Results against Dengler et al., 2021

Custom Dataset Experiments		
Dataset	**IoU**	**Accuracy**
Office 1 [1]	0.58	0.743
Office 2 [1]	0.51	0.636
Conference Room	0.806	0.64
Kitchen Room	0.804	0.651
Office Room	0.783	0.643
Experiments Station Room	0.809	0.631
Swarm Robots Room	0.801	0.645
TUM RGB-D Dataset Experiments		
fr2_desk [1]	0.567	**0.671**
fr3_sitting_xyz [1]	0.615	**0.652**
fr2_desk [ours]	**0.776**	0.651
fr3_sitting_xyz [ours]	**0.765**	0.645

We compare our mapping results against the reference work [1] by Dengler et al., 2021. As mentioned before, there were no major works or state-of-the-art results for semantic material mapping. So we compare our model performance at the semantic object level. The results of the reference work [1] by Dengler et al., 2021 is shown in Table 4.17. The reference work performance on the TUM RGB-D dataset is also presented in the table. Our model performs better than the reference work in terms of IoU and got comparable results in terms of Accuracy.

[1] [1] Dengler, Nils, et al. "Online object-oriented semantic mapping and map updating." 2021 European Conference on Mobile Robots (ECMR). IEEE, 2021.

CHAPTER 5

CONCLUSION

The problem of semantic material labeling of 3D point clouds has been an active area of research in computer vision and robotics. This involves assigning a semantic material label to each point in the 3D point cloud that represents the type of material in the physical world. This information is vital for a variety of applications such as robot navigation, object manipulation, and scene understanding.

To solve this problem, we propose an architecture composed of several components, including RGB-D sensor, Yolov5 object detection, material classification network, ORB-SLAM2, Voxel-based matching component, and MSCC algorithm (multi-scale connected components). This architecture aims to provide a real-time, web-based, and scalable solution for semantic annotation of 3D point cloud data. This framework also allows the integration of 2D and 3D data to provide a semantic understanding of the environment.

An RGB-D sensor is used to collect 3D point cloud data from the environment. The use of RGB-D data provides the system with rich and complementary information, allowing for highly accurate object and material discrimination. The RGB-D sensor captures the ambient depth and color information, which is then processed using ORB-SLAM2 to create a dense point cloud. The depth information provided by this sensor can also be used by ORB-SLAM2 to track the position of the sensor in real-time. Yolov5 object detection was used to identify objects in the scene and determine the region of interest for material classification. The material classification network then uses the region of interest to classify the materials in the scene. Using a material classification network further improves labeling accuracy as the system can classify materials based on their structure and appearance. This network was trained on a large dataset, allowing it to generalize well to new environments. ORB-SLAM2 is used to create a 3D map of the environment, which is further processed by a Voxel-based matching component. ORB-SLAM2 is a fast and efficient way to capture and plot a 3D point cloud in real-time, essential in dynamic and highly cluttered environments.

The Voxel-Based Matching Component and the MSCC algorithm play a crucial role in the system by allowing the material label to propagate and refine the segmentation results. The Voxel-Based Matching component is responsible for matching and registering the point clouds obtained from the RGB-D cameras. By clustering the point cloud into voxels, the system is able to effectively identify and segment individual objects in the scene, while the MSCC algorithm is used to connect voxels into regions and objects that are coherent in terms of their color, texture, and material properties. The use of the MSCC algorithm enables the system to perform material labeling at a faster rate compared to traditional algorithms. This is achieved by using a hierarchical approach to perform the labeling, which reduces the computational cost.

The results of the proposed architecture have shown that the system is able to obtain high accuracy in terms of material labeling and segmentation. The system is able to effectively differentiate between objects and materials in real-world scenes, together with objects made of one-of-a-kind materials such as wood, metal, glass, etc. In addition, the system is able to precisely propagate the material labels at some stage in the scene, even in tremendously cluttered environments the place objects are occluded or partly visible. Furthermore, we demonstrate that the proposed architecture presents a promising solution for semantic material labeling of 3D point clouds. The combination of various components and algorithms permits for a comprehensive approach to the problem, leading to accurate and efficient results. The use of the MSCC algorithm for material label propagation is a novel contribution to the field and has been shown to produce good results. However, further research is needed to address the limitations and improve the system.

It is really worth mentioning that the proposed architecture can also be easily extended to include additional features and capabilities. For example, the system can be modified to include additional material classes, enhance the accuracy of the material classification network, or incorporate additional components to enhance the overall performance of the system. Additionally, the system can also be tailored to quite a number of applications such as building information modeling, construction site monitoring, and robotics. The combination of different components and algorithms permits the system to perform semantic material labeling in real-time with high accuracy, making it a valuable tool for a variety of applications. The proposed structure offers a solid foundation for future work and research in the field of semantic material labeling of 3D factor clouds.

Overall, the proposed architecture represents a significant advancement in the field of semantic material labeling of 3D factor clouds. It gives a practical solution for various applications in the areas of robotics and computer vision and has the potential to make a significant impact in these fields. The ability to accurately and efficiently label the materials in a 3D point cloud offers a wealth of information for various applications, and the results achieved using this architecture display its usage for practical applications.

§ 5.1 Limitations and Future Work

§ 5.1.1 Limitations

However, the proposed architecture may have some limitations. For example, it may not perform well in cluttered scenes or scenes with highly reflective or transparent objects. Additionally, the current implementation of the Material Classification Network may not be able to classify all possible materials in the scene, leading to incomplete material labeling of the point cloud.

The accuracy of the material labeling depends on the accuracy of object detection, material classification, and 3D localization. Any errors in any of these components can propagate and affect the final results.

The performance of the system is highly dependent on the quality and density of the point cloud. The system may struggle to provide accurate results in situations where the point cloud is sparse or noisy.

The size of the 3D point cloud is a factor in the processing time of the system. The larger the point cloud, the longer it will take to process, potentially leading to real-time constraints for certain applications. The system requires a high computational cost, as it involves multiple algorithms and components that are computationally intensive.

The current system only labels the material of objects that have been detected by YOLOv5. Any objects that are missed by the object detector will not be labeled with their material type.

These limitations highlight the need for further research and development in the field to address these challenges and improve the robustness and accuracy of the system.

§ 5.1.2 Future Work

Future work can be progressed in several ways to enhance the proposed architecture. Some possible directions are:

Improving the object detection accuracy: The accuracy of object detection can be improved by fine-tuning the YOLOv5 network on a dataset that is more representative of the environment in which the system will be used.

Improving the material classification accuracy: The material classification network can be fine-tuned on a larger and more diverse dataset to improve its accuracy.

Incorporating additional modalities: The system can be extended to incorporate additional modalities, such as depth and thermal cameras, to improve the quality and density of the point cloud. Using additional sensors such as LiDAR or stereo cameras to provide additional information and improve the overall accuracy of the system.

Improving the efficiency of the system: The efficiency of the system can be improved by exploring parallel processing techniques and optimizing the algorithms used in the voxel-based matching component.

Overall, there is a lot of potentials for future work to enhance the proposed architecture and improve the results of the semantic material labeling system.

BIBLIOGRAPHY

Adams, R., & Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*(6), 641–647. https://doi.org/10.1109/34.295913

Bell, S., Upchurch, P., Snavely, N., & Bala, K. (2014). Material recognition in the wild with the materials in context database. *CoRR*, *abs/1412.0623*. http://arxiv.org/abs/1412.0623

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Campos, C., Elvira, R., Rodríguez, J. J. G., M. Montiel, J. M., & D. Tardós, J. (2021). *Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam*. https://doi.org/10.1109/TRO.2021.3075644

Chen, H., & Li, Y. (2018). Progressively complementarity-aware fusion network for rgb-d salient object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3051–3060.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., & Vedaldi, A. (2013). Describing textures in the wild. *CoRR*, *abs/1311.3618*. http://arxiv.org/abs/1311.3618

Couprie, C., Farabet, C., Najman, L., & LeCun, Y. (2013). Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*.

Cui, L., & Ma, C. (2020). Sdf-slam: Semantic depth filter slam for dynamic environments. *IEEE Access*, *8*, 95301–95311. https://doi.org/10.1109/ACCESS.2020.2994348

Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., & Theobalt, C. (2017). Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. (3). https://doi.org/10.1145/3054739

DeGol, J., Fard, M. G., & Hoiem, D. (2016). Geometry-informed material recognition. *CoRR*, *abs/1607.05338*. http://arxiv.org/abs/1607.05338

Dengler, N., Zaenker, T., Verdoja, F., & Bennewitz, M. (2021). Online object-oriented semantic mapping and map updating. *2021 European Conference on Mobile Robots (ECMR)*, 1–7. https://doi.org/10.1109/ECMR50962.2021.9568817

Desingh, K., Krishna, K. M., Rajan, D., & Jawahar, C. V. (2013). Depth really matters: Improving visual salient region detection with depth. *BMVC*.

Ding, Y., Liu, Z., Huang, M., Shi, R., & Wang, X. (2019). Depth-aware saliency detection using convolutional neural networks. *J. Vis. Comun. Image Represent.*, *61*(100), 1–9. https://doi.org/10.1016/j.jvcir.2019.03.019

Endres, F., Hess, J., Sturm, J., Cremers, D., & Burgard, W. (2014). 3D mapping with an RGB-D camera. *IEEE Transactions on Robotics*, *30*(1), 177–187.

Girshick, R. (2015). Fast r-cnn, 1440–1448. https://doi.org/10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, 580–587. https://doi.org/10.1109/CVPR.2014.81

Golovinskiy, A., & Funkhouser, T. (2009). Min-cut based segmentation of point clouds. *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, 39–46. https://doi.org/10.1109/ICCVW.2009.5457721

Han, J., Chen, H., Liu, N., Yan, C., & Li, X. (2018). Cnns-based rgb-d saliency detection via cross-view transfer and multiview fusion. *IEEE transactions on cybernetics*, *48 11*.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn, 2961–2969.

Hempel, T., & Al-Hamadi, A. (2022). An online semantic mapping system for extending and enhancing visual slam. *Engineering Applications of Artificial Intelligence*, *111*, 104830.

Hu, D., Bo, L., & Ren, X. (2011). Toward robust material recognition for everyday objects. *BMVC*.

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., & Markham, A. (2020). Randla-net: Efficient semantic segmentation of large-scale point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11108–11117.

Huang, J., Xie, L., Wang, W., Li, X., & Guo, R. (2022). a Multi-Scale Point Clouds Segmentation Method for Urban Scene Classification Using Region Growing Based on Multi-Resolution Supervoxels with Robust Neighborhood. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *43B5*, 79–86. https://doi.org/10.5194/isprs-archives-XLIII-B5-2022-79-2022

Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., TaoXie, Fang, J., imyhxy, Michael, K., Lorna, V, A., Montes, D., Nadar, J., Laughing, tkianai, yxNONG, Skalski, P., Wang, Z., … Minh, M. T. (2022). *ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference* (Version v6.1). Zenodo. https://doi.org/10.5281/zenodo.6222936

Kalliatakis, G., Stamatiadis, G., Ehsan, S., Leonardis, A., Gall, J., Sticlaru, A., & McDonald-Maier, K. D. (2017). Evaluating deep convolutional neural networks for material classification. *CoRR*, *abs/1703.04101*. http://arxiv.org/abs/1703.04101

Kim, J., Lim, H., Ahn, S. C., & Lee, S. (2018). Rgbd camera based material recognition via surface roughness estimation. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1963–1971. https://doi.org/10.1109/WACV.2018.00217

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. https://doi.org/10.1145/3065386

Lai, K., Bo, L., Ren, X., & Fox, D. (2011). A large-scale hierarchical multi-view rgb-d object dataset. *2011 IEEE International Conference on Robotics and Automation*, 1817–1824.

Lee, S., Lee, D., Kim, H.-C., & Lee, S. (2022). Material type recognition of indoor scenes via surface reflectance estimation. *IEEE Access*, *10*, 134–143. https://doi.org/10.1109/ACCESS.2021.3137585

Lee, S., Lim, H., Ahn, S., & Lee, S. (2019). Ir surface reflectance estimation and material type recognition using two-stream net and kinect camera. *ACM SIGGRAPH 2019 Posters*. https://doi.org/10.1145/3306214.3338557

Li, Y., Zhang, J., Cheng, Y., Huang, K., & Tan, T. (2018). Df 2 net: Discriminative feature learning and fusion network for rgb-d indoor scene classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1).

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., & Chen, B. (2018). Pointcnn: Convolution on x-transformed points. *arXiv preprint arXiv:1801.07791*.

Li, Z., & Zhou, F. (2017). Fssd: Feature fusion single shot multibox detector. *arXiv preprint arXiv:1712.00960*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755.

Liu, C., Sharan, L., Adelson, E. H., & Rosenholtz, R. (2010). Exploring features in a bayesian framework for material recognition. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 239–246. https://doi.org/10.1109/CVPR.2010.5540207

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector, 21–37.

Liu, W., & Tao, D. (2013). Multiview hessian regularization for image annotation. *IEEE Transactions on Image Processing*, *22*(7), 2676–2687. https://doi.org/10.1109/TIP.2013.2255302

Long, J., Shelhamer, E., & Darrell, T. (2014). Fully convolutional networks for semantic segmentation. *CoRR*, *abs/1411.4038*. http://arxiv.org/abs/1411.4038

McCormac, J., Handa, A., Davison, A., & Leutenegger, S. (2017). Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 4628–4635. https://doi.org/10.1109/ICRA.2017.7989538

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, *31*(5), 1147–1163.

Mur-Artal, R., & Tardós, J. D. (2017). ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, *33*(5), 1255–1262. https://doi.org/10.1109/TRO.2017.2705103

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, *30*.

Qi, X., Xiao, R., Li, C.-G., Qiao, Y., Guo, J., & Tang, X. (2014). Pairwise rotation invariant co-occurrence local binary pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(11), 2199–2213. https://doi.org/10.1109/TPAMI.2014.2316826

Rabbani, T., van den Heuvel, F., & Vosselman, G. (2006). Segmentation of point clouds using smoothness constraint.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection, 779–788.

Redmon, J., & Farhadi, A. (2017). Yolo9000: Better, faster, stronger, 6517–6525. https://doi.org/10.1109/CVPR.2017.690

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

Rusu, R. B., & Cousins, S. (2011). 3d is here: Point cloud library (pcl). *2011 IEEE International Conference on Robotics and Automation*. https://doi.org/10.1109/ICRA.2011.5980567

Salas-Moreno, R. F., Newcombe, R. A., Strasdat, H., Kelly, P. H., & Davison, A. J. (2013). Slam++: Simultaneous localisation and mapping at the level of objects. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 1352–1359. https://doi.org/10.1109/CVPR.2013.178

Schnabel, R., Wahl, R., & Klein, R. (2007). Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, *26*.

Schops, T., Sattler, T., & Pollefeys, M. (2019). Bad slam: Bundle adjusted direct rgb-d slam. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schwartz, G., & Nishino, K. (2013). Visual material traits: Recognizing per-pixel material context. *2013 IEEE International Conference on Computer Vision Workshops*, 883–890. https://doi.org/10.1109/ICCVW.2013.121

Schwartz, G., & Nishino, K. (2016). Material recognition from local appearance in global context. *CoRR*, *abs/1611.09394*. http://arxiv.org/abs/1611.09394

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Shafiee, M. J., Chywl, B., Li, F., & Wong, A. (2017). Fast yolo: A fast you only look once system for real-time embedded object detection in video. *ArXiv*, *abs/1709.05943*.

Song, S., Lichtenberg, S. P., & Xiao, J. (2015). Sun rgb-d: A rgb-d scene understanding benchmark suite. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Song, X., Jiang, S., & Herranz, L. (2017). Combining models from multiple sources for rgb-d scene recognition. *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 4523–4529.

Stein, S. C., Schoeler, M., Papon, J., & Wörgötter, F. (2014). Object partitioning using local convexity. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 304–311.

Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection. *NIPS*.

Tateno, K., Tombari, F., Laina, I., & Navab, N. (2017). Cnn-slam: Real-time dense monocular slam with learned depth prediction. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6565–6574. https://doi.org/10.1109/CVPR.2017.695

Trevor, A. J. B., Gedikli, S., Rusu, R. B., & Christensen, H. I. (2013). Efficient organized point cloud segmentation with connected components.

Wang, A., Cai, J., Lu, J., & Cham, T.-J. (2015). Mmss: Multi-modal sharable and specific feature learning for rgb-d object recognition. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1125–1133. https://doi.org/10.1109/ICCV.2015.134

Wang, N., & Gong, X. (2019). Adaptive fusion for RGB-D salient object detection. *CoRR*, *abs/1901.01369*. http://arxiv.org/abs/1901.01369

Wang, Q., Chen, M., Nie, F., & Li, X. (2020). Detecting coherent groups in crowd scenes by multiview clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(1), 46–58. https://doi.org/10.1109/TPAMI.2018.2875002

Wang, T., Zhu, J., Ebi, H., Chandraker, M., Efros, A. A., & Ramamoorthi, R. (2016). A 4d light-field dataset and CNN architectures for material recognition. *CoRR*, *abs/1608.06985*. http://arxiv.org/abs/1608.06985

Wu, B., Liu, Y., Lang, B., & Huang, L. (2018). Dgcnn: Disordered graph convolutional neural network based on the gaussian mixture model. *Neurocomputing*, *321*, 346–356.

Xiang, Y., & Fox, D. (2017). Da-rnn: Semantic mapping with data associated recurrent neural networks. *arXiv preprint arXiv:1703.03098*.

Yu, C., Liu, Z., Liu, X.-J., Xie, F., Yang, Y., Wei, Q., & Fei, Q. (2018). Ds-slam: A semantic visual slam towards dynamic environments. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1168–1174.

Yu, J., Rui, Y., & Tao, D. (2014). Click prediction for web image reranking using multimodal sparse coding. *IEEE Transactions on Image Processing*, *23*(5), 2019–2032.

Yu, J., Yang, X., Gao, F., & Tao, D. (2016). Deep multimodal distance metric learning using click constraints for image ranking. *IEEE transactions on cybernetics*, *47*(12), 4014–4024.

Zhao, C., Sun, L., & Stolkin, R. (2017). A fully end-to-end deep learning approach for real-time simultaneous 3d reconstruction and material recognition. *2017 18th International Conference on Advanced Robotics (ICAR)*, 75–82.